Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Tomi-Mikael Kahilakoski

Empirical Comparison of Canonical Correlation Analysis and its Kernelized Variants

Master's Thesis
Espoo, July 24, 2019

Supervisor: Professor Juho Rousu
Advisor: Viivi Uurtio M.Sc. (Tech.)

Aalto University
School of Science

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Tomi-Mikael Kahilakoski		
Title:	Empirical Comparison of Canonical Correlation Analysis and its Kernelized Variants		
Date:	July 24, 2019	**Pages:**	60
Major:	Computer Science	**Code:**	SCI3042
Supervisor:	Professor Juho Rousu		
Advisor:	Viivi Uurtio M.Sc. (Tech.)		

Finding non-linear relationships in data sets has become important in many fields of science. We compared the performances of CCA and three dense, kernelized CCA variants, KCCA, gradKCCA, (S)CCA-HSIC in 12 experiments with simulated data and two real-world scenarios. We concluded that the kernelized CCA variants are capable of retrieving complex non-linear relationships in simulated settings, and, in agreement with previous research, the methods can be applied in real-world scenarios.

Keywords:	CCA, KCCA, gradKCCA, (S)CCA-HSIC, kernelized variant
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Tomi-Mikael Kahilakoski		
Työn nimi:	Kanonisen korrelaatioanalyysin ja sen ydinfunktiolaajennusten empiirinen vertailu		
Päiväys:	24. heinäkuuta 2019	**Sivumäärä:**	60
Pääaine:	Tietotekniikka	**Koodi:**	SCI3042
Valvoja:	Professori Juho Rousu		
Ohjaaja:	Diplomi-insinööri Viivi Uurtio		

Epälineaaristen yhteyksien löytämisestä on tullut tärkeää monilla tieteenaloilla. Tutkimuksessa vertailin toisiinsa CCA:a sekä kolmea CCA:n tiheää ydinfunktiolaajennusta: KCCA:a, gradKCCA:a ja (S)CCA-HSIC:a. Tutkimuksen ensimmäisessä osassa loin 12 koeasetelmaa, joissa käytin simuloitua dataa. Toisessa osassa vertailin menetelmien kykyä löytää yhteyksiä kahdella tosimaailman datajoukolla. Tutkimuksessa ilmeni, että ydinfunktiolaajennetut CCA-muunnelmat kykenevät löytämään monimutkaisia epälineaarisia yhteyksiä simuloiduissa asetelmissa. Aiempaa tutkimusta vahvistaen havaitsin, että menetelmät soveltuvat myös tosimaailman datan kanssa käytettäväksi.

Asiasanat:	CCA, KCCA, gradKCCA, (S)CCA-HSIC, ydinfunktiolaajennus
Kieli:	Englanti

Acknowledgements

I want to thank my supervisor, Professor Juho Rousu and advisor, M.Sc. Viivi Uurtio for their assistance and advice during the writing of this thesis. I also want to thank Olli-Pekka Kahilakoski for proof-reading, and my family and friends for their support.

Espoo, July 24, 2019

Tomi-Mikael Kahilakoski

Abbreviations and Acronyms

Operators

ξ	noise term
u, v	weight vectors in the input space
$\phi(x) : X \to F$	mapping to feature space
$\tilde{A}$, $\tilde{v}$	centered matrix A or vector v
$\langle x, z \rangle$	inner product between x and z
trace(A)	trace operation of matrix A or the sum of the elements on the main diagonal
$\|v\|_p$	p norm
e	base of the natural logarithm
$A^\intercal$	transpose of matrix A
$\frac{\mathrm{d}}{\mathrm{d}t}$	derivative with respect to variable t
$\frac{\partial}{\partial t}$	partial derivative with respect to variable t
$\sum_i$	sum over index i
ρ	Lagrange multiplier
L	primal Lagrangian
N	number of observations

Abbreviations

CCA	canonical correlation analysis
RKHS	reproducing kernel Hilbert space
KCCA	kernelized CCA
HSIC	Hilbert-Schmidt independence criterion

Contents

Chapter 1

Introduction

In many fields of science, it has become necessary to discover nonlinear relationships in data. Discovering non-linear relationships is generally not possible using linear methods, so non-linear methods are needed. This thesis focuses on a family of methods capable of learning the relationships in multivariate data, that is canonical correlation analysis (CCA) and its kernelized variants.

Canonical correlation analysis is a linear method capable of discovering linear associations only. For the purpose of finding non-linear relationships, kernel methods need to be applied to CCA. Kernel methods can be used for modifying a linear method into non-linear one, and they have already been used in many areas of data science and machine learning. Since the early 2000s, numerous kernelized CCA variants have been proposed, and they have become a well-established tool for discovering non-linear relationships within multivariate data settings [1] [2] [3].

Several kernelized CCA variants have been proposed along the years, all of which have their strengths and weaknesses. Studying the capabilities and limitations of the methods is crucial for knowing which method is suitable for the problem at hand. Scalability, robustness to noise variables, the capability to discover complex non-linear relationships, robustness against outliers in data are some of the issues that have been mentioned in the research field of kernelized CCA variants.

Canonical correlation analysis (CCA), which is one of the four methods studied in this thesis, discovers linear interrelations between two divisions of a data set. It was originally proposed by Hotelling in 1936 [4]. Kernel CCA (KCCA), which is another variant studied in this paper, was the earliest of the kernelized CCA variants. It was proposed by Akaho in 2001 [1]. The lack of sparsity, difficulty in interpreting the result and the incapability to scale well to settings with large number of observations has advanced the development

of other kernelized CCA variants. In this thesis, we also study (S)CCA-HSIC [2], which uses Hilbert-Schmidt independence criterion (HSIC) [5] as the measure of dependence instead of correlation. It has certain advantages in terms of computational requirements compared to KCCA and is also easier to interpret due to having access to the canonical weight vectors in the original input space. In Uurtio et al. [3], (S)CCA-HSIC has also been found to extract complex non-linear relationships better than KCCA. Another method studied in this thesis is gradient-based kernel canonical correlation analysis (gradKCCA), which was proposed in 2019 [6]. GradKCCA was designed to be a less computationally intensive alternative to (S)CCA-HSIC, yet it has the interpretability of (S)CCA-HSIC.

In this thesis, we aim to empirically evaluate the performances of the four CCA methods, CCA, KCCA, gradKCCA and (S)CCA-HSIC in linear and non-linear settings using simulated and real-world scenarios. We run 12 experiments on simulated data, aiming to find out what sort of relationships each method is capable of discovering and where the limits are for the complexity of the relationships. In real-world experiments, we use two publicly available data sets to evaluate how well each method performs in real-world settings. We expect to find differences in the performances of the methods. After running simulated experiments and real-world data experiments, we conclude with the strengths and the weaknesses of each method.

Kernel methods are based on kernel functions. The ones that we use in this thesis are linear, polynomial and Gaussian kernels. For KCCA, we use a linear and polynomial kernel. For gradKCCA, we use polynomial and Gaussian kernels. For HSIC-CCA, a universal kernel is required, so we only use a Gaussian kernel.

While sparse models excel in settings with high number of variables, we limit the scope of this thesis to dense models. Therefore, the l_2 regularization term in CCA-HSIC and gradKCCA is used. As for CCA and KCCA, the models are dense.

In this thesis, we are looking at the capability of the CCA variants to extract complicated, non-linear as well as linear relationships in multivariate settings. We cover settings with low numbers of observations and low amount of noise. In this thesis, we do not investigate the speed or scalability of the methods.

Chapter 2

Background

2.1 Basic statistical concepts

Statistics is a methodology for collecting, analyzing, interpreting and drawing conclusions from data [7]. When a researcher has a question about some data, statistics aims to answer how to gather and analyze the data in a way that is relevant to the question. It offers methods to summarize and display data to get answers that are supported by the data [8]. Statistics has two philosophical approaches, the frequentist approach and the Bayesian approach. The main difference between the two approaches is in the interpretation of probability.

Random variable is a variable in statistics, the values of which depend on a random phenomenon. A random variable is described by a probability distribution. For continuous random variable X, the expected value is defined as

$$\mathrm{E}(X) = \int_{-\infty}^{+\infty} x f(x)\, dx,$$

where $f(x)$ is the probability density function of X. The sample mean estimates the expected value of the distribution. Sample mean is defined as

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^{N} x_i.$$

For a discrete random variable, the expected value is calculated as

$$\mathrm{E}(X) = \sum_{i=1}^{n} P(X_i) X_i.$$

Variance is a measure of the variability of a random variable. Variance of random variable is defined as

$$\mathrm{Var}(X) = \mathrm{E}([X - \mathrm{E}(X)]^2),$$

where $\mathrm{E}(X)$ is the expected value of the random variable X. The variance is estimated by sample variance.

Covariance is a measure for joint variability of two random variables. For continuous random variables X and Y, covariance is defined as

$$\begin{aligned} \mathrm{Cov}(X, Y) &= \mathrm{E}((X - \mathrm{E}(X))(Y - \mathrm{E}(Y))) \\ &= \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y) \end{aligned} \tag{2.1}$$

The latter expression for covariance is derived from the definition of covariance by using the rules for additivity and multiplication by a constant for expected value.

2.2 Correlation, dependence and independence

Correlation is a measure of linear association between a pair of variables. The early work regarding the understanding of correlation was based on the research by Galton in 1885 [9] and Pearson in 1896 [10]. As a result, Pearson's correlation coefficient was born, which is the most basic measure of bivariate relationship [11]. Correlation was the first method for measuring the association between two variables without a cause-effect relationship. Since the creation, correlation coefficient has become a central statistical method for observational experiments in many disciplines, and is used in many areas of statistics and data science, including canonical correlation analysis [12] [13] [11] [4]. For variables X and Y, Pearson's correlation coefficient is defined as follows:

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}, \tag{2.2}$$

where σ_X and σ_Y are the standard deviations of X and Y. For a sample, the correlation coefficient is

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n s_x s_y},$$

where x and y are variables with N observations, s_x and s_y are the sample standard deviations of x and y and $\bar{x}$ and $\bar{y}$ are sample means of x and y, respectively. The numerator divided by n is equal to the empirical covariance between the two variables.

As can be concluded from (2.2), covariance also measures linear association between variables. However, its value is unbounded and depends on the scales of measurement for X and Y, often making it a suboptimal measure for linear association. When covariance is divided by the standard deviations of the two variables, it becomes the correlation coefficient, which then receives a value between 1 and -1. By obtaining a scaled value, it is easier to compare to the measures of other bivariate analyses [11].

When the Pearson correlation coefficient is positive, the variables are said to have a positive correlation. Likewise, variables have a negative correlation (or inverse correlation) when the value is negative. Strength of the relationship is measured by how close the correlation coefficient is to 1 or -1. A correlation coefficient value of 0 indicates that there is no correlation between the variables. In Figure 2.1, there are plots of variables with correlations of different strength and signs. As Figure 2.1 shows, a high correlation generally means that the points are located on a straight line. However, as Kotz et al. [12] show in their paper, an increasing correlation does not necessarily mean that the plots become more like a straight line.

During the early days of correlation research, correlation was incorrectly implied to be a measure of dependence [12]. However, correlation and dependence are not equivalent. Correlation coefficient only measures the degree of linear relationship, whereas dependence also accounts for non-linear relationships. In other words, there can be a dependence between two uncorrelated variables, but there can not be correlation without dependence. Two variables are independent if

$$f_X(x)f_Y(y) = f_{X,Y}(x, y), \tag{2.3}$$

where $f_X(x)$ and $f_Y(y)$ are the probability density functions of continuous random variables X and Y, and $f_{X,Y}(x, y)$ is the joint probability distribution of X and Y. Two variables can be uncorrelated yet still have a dependence. In [12] there is a simple example of this. Let us define the distributions of two discrete random variables,

$$P(X = Y = 0) = P(X = Y = 1) = P(X = -1, Y = 1) = \tfrac{1}{3},$$

Then, we can determine the marginal distributions for X and Y:

$$P(X = 0) = P(X = 1) = P(X = -1) = \tfrac{1}{3}$$
$$P(Y = 0) = \tfrac{1}{3}; P(Y = 1) = \tfrac{2}{3}$$

For continuous random variables the condition for independence is shown in (2.3). For discrete random variables, the equivalent condition for independence is $\forall x, y : P(X = x)P(Y = y) = P(X = x, Y = y)$. This condition

Table 2.1: The marginal distributions and the joint distribution for random variables X and Y. The marginal distribution for X is the 4th row in the table, and for Y it is the 5th column. The joint distribution is the 3×2 area in the middle.

	X=-1	X=0	X=1	
Y=0	0	1/3	0	1/3
Y=1	1/3	0	1/3	2/3
	1/3	1/3	1/3	

can be verified to not be true from Table 2.1. There, we can see that the condition for independence does not hold, because e.g. $P(X = 0)P(Y = 0) = \frac{1}{3} \times \frac{1}{3} \neq P(X = 0, Y = 0) = \frac{1}{3}$. Therefore, X and Y are dependent.
The correlation coefficient equals 0, if $\mathrm{Cov}(X, Y) = 0$. To calculate the covariance between X and Y, we first need to obtain the expected values

$$\mathrm{E}(X) = \frac{1}{3}(-1 + 0 + 1) = 0$$

$$\mathrm{E}(Y) = \frac{1}{3} \times 0 + \frac{2}{3} \times 1 = \frac{2}{3}$$

$$\mathrm{E}(XY) = \frac{1}{3}(0 \times 0 + 1 \times 1 + (-1 \times 1)) = 0.$$

Then, the covariance between X and Y can be calculated using (2.1):

$$\mathrm{Cov}(X, Y) = \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y) = 0 - 0 \times \frac{2}{3} = 0$$

Since the covariance is 0, the correlation coefficient is also 0. Therefore, between X and Y, there is dependence but no correlation.

Pearson's correlation coefficient has been criticized for its lack of obvious interpretation for its units [11]. For better interpretability, there exists coefficient of determination, the units of which can be interpreted as the proportion of the variation in the first variable that can be explained by the other variable. Coefficient of determination can be calculated by simply squaring the correlation coefficient. The value ranges from 0 to 1 and can be interpreted as a percentage.

Non-linear relationships have dependence but not necessarily correlation. Non-linear relationships can be divided into monotonic and non-monotonic relationships. The relationship is called monotonic when the shape of a relationship follows the shape of a monotonic function. For increasing monotonic functions it holds that if $x_1 < x_2$, then $f(x_1) \leq f(x_2)$. For decreasing monotonic functions it holds that if $x_1 < x_2$, then $f(x_1) \geq f(x_2)$.

Non-linear relationships exist in everyday life. For example, the more there are workers working on a given task, the less an additional worker shortens the time required to finish the task. Because examples can be found in everyday life, it is not surprising that non-linear relationships are also found in sciences. In medicine, because of the complex nature of human systems, non-linear behavior is common. Nevertheless, it has been common for physiologists and physicians to apply linear models to data analysis problems. Since physiologists and physicians have become more aware of the existence of non-linear relationships, non-linear models have become increasingly common. While linear modeling has been widely used in medicine, non-linear models have further extended the understanding of complex systems in medicine [14].

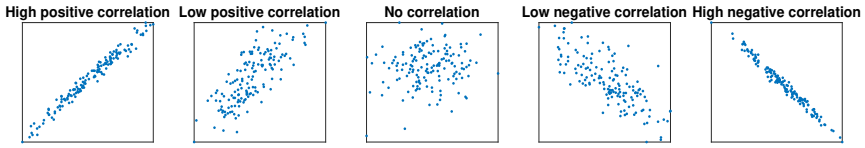


Figure 2.1: Plots of relationships of high and low, positive and negative correlations and no correlation. First on the left is a highly correlating positive relationship, the points of which are located on a line. The negative counterpart to that is in the rightmost plot. Second plot on the left has a lower correlating relationship, which is no more a line. Still, values of Y tend to increase as X increases. The corresponding relationship with negative correlation is the plot second from the right. Last, the no correlation plot in the middle shows how values of Y are not dependent on X.

2.3 Canonical correlation analysis

Canonical Correlation Analysis (CCA) is a multivariate analysis procedure for investigating relationships between two sets of variables, first proposed by Hotelling in 1935 [15]. It finds such linear combinations of variables in first and second variable sets that maximize the correlation between the two sets. The standard CCA has been extended to settings where there are too few observations with regard to the amount of variables, where the relations are non-linear or where the dimensionality of the data is too large for human interpretation [13].

Let us denote two data sets with variable numbers p and q and N observations, $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times q}$. The principal idea of CCA is to find such

linear combinations of variables of data sets X and Y that correlate with each other. The variables of X and Y are combined using canonical weights for each set in the following manner:

$$Xu = z_X, Yv = z_Y$$

where $u \in \mathbb{R}^p$ and $v \in \mathbb{R}^q$ are canonical weights and z_X and z_Y are canonical variables representing linear combinations of the variables of X and Y, respectively. Then, we inspect the correlation between the two canonical variables. CCA is looking for such canonical weights that produce transformed variables with the highest correlation. Therefore, the correlation is maximized:

$$\max \frac{\langle Xu, Yv \rangle}{\|Xu\|_2 \|Xv\|_2}$$

To limit the number of solutions to exactly one and at the same time get more simple function to maximize, we add a normalization constraint to z_X and z_Y:

$$\max \langle z_X, z_Y \rangle = \langle Xu, Yv \rangle$$

$$s.t. \|z_X\|_2 = 1, \|z_Y\|_2 = 1$$

CCA can be solved in multiple different ways, e.g. using the singular value decomposition or generalized eigenvalue problem. Here, we will show how to solve CCA through the standard eigenvalue problem. Such a maximizing problem can be solved using Lagrange multipliers.

As a part of solving the maximizing problem, we want to rewrite $\langle z_X, z_Y \rangle = z_X^\intercal z_Y = u^\intercal X^\intercal Y v$ in terms of sample covariance matrices between each other and itself $C_{XX} = \frac{1}{n-1} X^\intercal X; C_{XY} = \frac{1}{n-1} X^\intercal Y; C_{YY} = \frac{1}{n-1} Y^\intercal Y; C_{YX} = \frac{1}{n-1} Y^\intercal X$ and then substitute that into the function to be maximized:

$$\max \langle Xu, Yv \rangle = u^\intercal X^\intercal Y v = u^\intercal C_{XY} v$$

$$s.t. \|z_X\|_2 = \sqrt{u^\intercal C_{XX} u} = 1, \|z_Y\|_2 = \sqrt{v^\intercal C_{YY} v} = 1,$$

which can be solved using the Lagrangian multiplier method:

$$\mathcal{L} = u^\intercal C_{XY} v - \frac{\rho_1}{2}(u^\intercal C_{XX} u - 1) - \frac{\rho_2}{2}(v^\intercal C_{YY} v - 1),$$

where ρ_1 and ρ_2 are the Lagrange multipliers. Next, we take partial derivatives with respect to u and v,

$$\frac{\partial \mathcal{L}}{\partial u} = C_{XY} v - \rho_1 C_{XX} u = \bar{0} \tag{2.4}$$

$$\frac{\partial \mathcal{L}}{\partial v} = C_{YX}u - \rho_2 C_{YY}v = \bar{0} \tag{2.5}$$

Then the first equation is multiplied by $u^\intercal$ and the second equation by $v^\intercal$ from the left:

$$u^\intercal C_{XY}v - \rho_1 u^\intercal C_{XX}u = \bar{0}$$

$$v^\intercal C_{YX}u - \rho_2 v^\intercal C_{YY}v = \bar{0}$$

Since $u^\intercal C_{XX}u = 1$ and $v^\intercal C_{YY}v = 1$, we obtain

$$\rho_1 = \rho_2 = \rho$$

When (2.4) is solved with regard to vector u, we obtain

$$u = \frac{C_{XX}^{-1}C_{XY}v}{\rho} \tag{2.6}$$

By substituting this into (2.5), we obtain

$$\frac{1}{\rho}C_{YX}C_{XX}^{-1}C_{XY}v - \rho C_{YY}v = 0,$$

which can be formulated as a standard eigenvalue problem

$$C_{XX}^{-1}C_{YX}C_{YY}^{-1}C_{XY}v = \rho^2 v \tag{2.7}$$

The eigenvectors correspond to the weight vector v and then, weight vector u can be calculated using (2.6). Eigenvalues are equal to the square of the amount of correlation between the images constructed by the respective weight vectors. Thus, the correlation coefficients can be obtained by taking a square root of the eigenvalues.

In (2.7), the inverses of matrices C_{XX} and C_{YY} add prerequisites of the matrices being invertible. A small sample size, for example having less observations than variables in data sets, can cause the sample covariance matrices to not be invertible [13]. If the covariance matrices are not invertible, regularisation can be performed to make the covariance matrices invertible. Regularisation is done by adding a small positive constant to the diagonal of to matrices: $C_{XX} \hat{=} C_{XX} + cI$ (likewise for C_{YY}), where c is the regularisation constant and $I \in \mathbb{R}^{p \times p}$ is an identity matrix. Regularisation constants can be found automatically by using leave-one-out crossvalidation suggested by Leurgans et al. in 1993 [16].

The number of relationships that CCA retrieves is equal to the rank of $C_{XX}^{-1}C_{YX}C_{YY}^{-1}C_{XY}$ matrix. Due to CCA weight vectors being the eigenvectors of a single matrix, all relationships that CCA retrieves result in being

orthogonal to each other. Often, we are looking at a few weight vector pairs with the highest correlation.

CCA being a multivariate method, it has certain advantages. First, it has a lower risk of committing Type I error within a study. Type I error means that a method measures a false positive result, whereas Type II error means that a method measures a false negative result. This decreased likelihood of committing Type I error is due to having only a need to run CCA once for a given multivariate data set, instead of repeating a single-variate method for each variable. Another advantage of CCA is that many real-life phenomena have multiple causes and multiple effects, which multivariate techniques such as CCA are able to capture. For example, in psychology, if human behavior research is based on separately examining variables and their causes and effects, it may distort the complexity of the human behavior and cognitive systems. Last, CCA specifically is a comprehensive technique which can be used in various instances instead of other parametric tests [17].

The interpretation of CCA results is considered a rather difficult for beginning researchers to understand, which has been argued to be one reason for the somewhat limited utilization of CCA in the past [18]. In fact, the most general way of interpreting CCA results could be formulated as, "To what extent can one set of two or more variables be predicted or explained by another set of two or more variables?" [17]. CCA being a linear method, the strength of the weights tell how much they linearly contribute to the whole relationship.

CCA is a dense method, meaning that there is no penalty in the maximized formula for using more coefficients in the canonical weights. As such, with data sets with thousands of variables, it can become impossible to interpret the result. For better interpretation in that situation, a sparse model is needed.

2.4 Kernelized CCA variants

Since regular CCA is a linear technique, it can only retrieve relationships that are linear. To investigate systems with non-linear relationships, non-linear methods are needed. In this thesis, we will focus on kernelized CCA variants. In addition to kernel methods, also neural networks (Deep CCA by Andrew et al. 2013) and optimal scaling method by Burg and Leeuw (1983) have been succesfully used to retrieve non-linear relationships with CCA.

Kernel methods are based on kernel functions and reproducing kernel Hilbert spaces (RKHS), and can be used to make various linear machine learning methods non-linear. The idea of kernel methods is to use kernel

functions to map the data vectors into a higher-dimensional Hilbert space, also called feature space. Using the mapped data vectors, the algorithms are capable of working non-linearly.

Kernel methods use a mapping from the original space, input space, to Hilbert space, $\phi : X \to H, x \to \phi(x)$. The vectors in Hilbert space are of higher dimensionality than in input space, possibly infinite-dimensional. Then, to avoid an explicit expression for the non-linear mapping, Mercel's condition is applied. We can obtain inner products of data vector pairs in the Hilbert space by using so-called kernel trick. The kernel trick can be used whenever the algorithm can be written parameterized with inner products of the transformed data [19]. Having only the inner products is enough to work in the Hilbert space. A kernel function calculates the inner products of two data vectors in Hilbert space: $k(x, x') = \langle \phi(x), \phi(x') \rangle$, where ϕ represents the mapping function into feature space. Kernel matrix, or Gram matrix, is a $N \times N$ matrix with the inner products of each vector pair:

$$K = \begin{bmatrix} K_{x_1,x_1} & K_{x_1,x_2} & \dots & K_{x_1,x_N} \\ K_{x_2,x_1} & K_{x_2,x_2} & \dots & K_{x_2,x_N} \\ \vdots & \vdots & \ddots & \vdots \\ K_{x_N,x_1} & K_{x_N,x_2} & \dots & K_{x_N,x_N} \end{bmatrix},$$

where $K_{x_i,x_j} = \langle \phi(x), \phi(x') \rangle$ [20].

There are numerous kernel functions. However, the ones used in this thesis are:

$$\text{Linear kernel:} \quad K(x, x') = x^\intercal x' + c$$

$$\text{Polynomial kernel:} \quad K(x, x') = (\alpha x^\intercal x' + c)^d$$

$$\text{Gaussian kernel:} \quad K(x, x') = \exp(-\tfrac{\|x-x'\|^2}{2\sigma^2})$$

Kernel methods have become an essential tool in machine learning and pattern recognition. There are three reasons for this. First, there is a need for non-linear models, and unlike neural networks, kernel methods allow the translation of linear methods into non-linear methods while still working with linear algebra. Second, the kernel trick gives flexibility to uncover complicated non-linear relations. Third reason is that kernelized algorithms are less likely to overfit than other non-linear learning algorithms due to typically having fewer parameters and a more natural regularisation. Overall, kernel methods have been widely adopted in many fields, including computer vision, time-series analysis, physical sciences, and signal and image processing [20].

Unlike in CCA, where the coefficients in canonical weights indicate a monotonic relationship, the coefficients in non-linear methods do not allow

for identifying the type of the relationship. Instead, they only indicate the variables in the two views that are related [2].

The first propositions to extend CCA into using kernel methods were presented in [1], [19],[21], [22] and [23]. Since then, numerous kernelized variants have been developed for CCA. The kernelized CCA variants used in this thesis are KCCA, GradKCCA and (S)CCA-HSIC, which we will introduce in the following chapters.

2.4.1 Kernel CCA

Kernel CCA (KCCA) is a dense, kernelized version of CCA. In KCCA, instead of using the original data matrices $X \in \mathbb{R}^{N\times p}$ and $Y \in \mathbb{R}^{N\times q}$, we use the kernel matrices $K_X \in \mathbb{R}^{N\times N}$ and $K_Y \in \mathbb{R}^{N\times N}$, which consist of pairwise inner products of the original data vectors in Hilbert space [21]. The canonical weights in the feature space we denote with $\alpha \in \mathbb{R}^N$ and $\beta \in \mathbb{R}^N$. Similar to regular CCA, in KCCA, we are maximizing the correlation between the two canonical variables. Therefore, the kernelized CCA is formulated as

$$\max_{z_a, z_b \in \mathbb{R}^n} \langle z_a, z_b \rangle = \alpha^T K_X^T K_Y \beta$$

$$s.t. \|z_X\|_2 = \sqrt{\alpha^\intercal K_X^2 \alpha} = 1, \|z_Y\|_2 = \sqrt{\beta^\intercal K_Y^2 \beta} = 1$$

The maximizing problem of KCCA can be solved using Lagrangian multipliers

$$\mathcal{L} = \alpha^\intercal K_X^\intercal K_Y \beta - \frac{\rho_1}{2}(\alpha^\intercal K_X^2 \alpha - 1) - \frac{\rho_2}{2}(\beta^\intercal K_Y^2 \beta - 1),$$

where ρ_1 and ρ_2 are the Lagrange multipliers. Next, we take partial derivatives with respect to α and β,

$$\frac{\partial \mathcal{L}}{\partial \alpha} = K_X K_Y \beta - \rho_1 K_X^2 \alpha = \bar{0} \tag{2.8}$$

$$\frac{\partial \mathcal{L}}{\partial \beta} = K_Y K_X \alpha - \rho_2 K_Y^2 \beta = \bar{0} \tag{2.9}$$

Then, the first equation is multiplied by $\alpha^\intercal$ and the second equation by $\beta^\intercal$ from the left

$$\alpha^\intercal K_X K_Y \beta - \rho_1 \alpha^\intercal K_X^2 \alpha = \bar{0}$$

$$\beta^\intercal K_Y K_X \alpha - \rho_2 \beta^\intercal K_Y^2 \beta = \bar{0}$$

Because $\alpha^\intercal K_X^2 \alpha = 1$ and $\beta^\intercal K_Y^2 \beta = 1$, we obtain

$$\rho_1 = \rho_2 = \rho$$

Then, we solve α in (2.8)

$$\alpha = \frac{K_X^{-1} K_X^{-1} K_X K_Y \beta}{\rho} = \frac{K_X^{-1} K_Y \beta}{\rho} \tag{2.10}$$

Then, we substitute α into (2.9) and obtain

$$\frac{1}{\rho} K_Y K_X K_X^{-1} K_Y \beta - \rho K_Y^2 \beta = 0$$

By moving some of the terms, we obtain a standard eigenvalue problem,

$$K_Y^2 \beta = \rho^2 K_Y^2 \beta,$$

which reduces to

$$I\beta = \rho^2 \beta$$

This eigenvalue problem only has a trivial answer of $\rho^2 = 1$. Through regularisation of the kernel matrices, nontrivial solutions can be obtained.

$$\alpha^\intercal (K_X + c_1 I)^2 \alpha = 1$$

$$\beta^\intercal (K_Y + c_2 I)^2 \beta = 1$$

Then, the standard eigenvalue problem becomes

$$(K_Y + c_1 I)^{-2} K_Y K_X (K_X + c_2 I)^{-2} K_X K_Y \alpha = \rho^2 \alpha$$

After solving α through standard eigenvalue problem, β can be solved from (2.10). With regular CCA, the canonical weights u and v tell us which variables constitute to the correlation found by the method. In Kernel CCA, however, since the canonical weights $\alpha \in \mathbb{R}^N$ and $\beta \in \mathbb{R}^N$ are not weights for the data in the input space, they do not provide much insight to the relationships between sets X and Y [2]. Hardoon et al. [24] have suggested to retrieve the approximations of corresponding canonical weights u and v in the input space by $\tilde{u} = X_{train}^\intercal \alpha$ and $\tilde{v} = Y_{train}^\intercal \beta$.

In KCCA, the dimensionality of matrices K_X, K_Y is $\mathbb{R}^{N \times N}$, which causes the complexity of KCCA to be quadratic [6]. In 2004, Hardoon et al. proposed Incomplete Cholesky decomposition and Partial Gram-Schmidt orthogonalization (PGSO) to speed up the computations [24].

2.4.2 GradKCCA

GradKCCA is a large-scale sparse non-linear CCA method. It uses kernel functions similar to KCCA but does not need a kernel matrix in its implementation. Therefore, it does not suffer from the scaling problem that algorithms using kernel matrices suffer [6].

GradKCCA is based on KCCA. It adds a constraint that the canonical weights in the feature space have pre-images in the input space. This can be expressed as $\phi(u) = w_X$, $\phi(v) = w_Y$. There are two benefits to this. First, we get to work in the input space and obtain canonical weights corresponding to the input space. Second, a l_1 norm constraint can be set on the pre-images. Neither of these advantages are available in KCCA.

With the constraint $\phi(u) = w_X$, $\phi(v) = w_Y$, the optimization problem can be written as

$$\rho = \max_{u,v} \frac{\sum_{i=1}^{N} \langle \phi_Y(x_i), \phi_X(u) \rangle \langle \phi_Y(y_i), \phi_Y(v) \rangle}{\|(\langle \phi_X(x_i), \phi_X(u) \rangle)_{i=1}^{N}\| \|(\langle \phi_Y(y_i), \phi_Y(v) \rangle)_{i=1}^{N}\|},$$

where $u \in \mathbb{R}^p, v \in \mathbb{R}^q$ are the canonical weights in input space and N is the number of observations. Since the inner products can be written as a vectors of kernel function values $k^X(u) = (k^X(x_i, u))_{i=1}^{N}$ and $k^Y(v) = (k^Y(x_i, v))_{i=1}^{N}$, we obtain

$$\rho = \max_{u,v} \frac{k^X(u)^\intercal k^Y(v)}{\|k^X(u)\| \|k^Y(u)\|}$$

As is evident from the formulation, the vectors $k^X(u)$ and $k^Y(u)$ have N elements. Computationally, this translates into complexity $O(N)$, whereas KCCA, which computes kernel matrices of size $N \times N$, has a quadratic complexity $O(N^2)$. The difference in complexity has been found to cause a difference in speed. Uurtio et al. have shown that gradKCCA can improve computation speed and robustness to noise compared to other non-linear CCA methods, such as KCCA [6].

2.4.3 (S)CCA-HSIC

(S)CCA-HSIC is a kernelized CCA variant for finding sparse non-linear relationships by maximizing Hilbert-Schmidt independence criterion (HSIC). It is based on CCA-HSIC, which was originally proposed by Chang et al. in 2013 [2]. (S)CCA-HSIC was proposed due to the lack of sparsity of CCA-HSIC, which can make interpretation of results difficult when the dimensionality of the data is high [3].

Whereas previous CCA variants maximize correlation, (S)CCA-HSIC uses Hilbert-Schmidt independence criterion (HSIC) as a measure of dependence

that is maximized. HSIC can be used as a measure of dependence as long as it is associated with a universal kernel [5]. Gretton et al. have proved that HSIC value is zero if and only if the variables are independent [5].

The dependence between two sets of variables is defined as the squared Hilbert-Schmidt norm of the associated cross-covariance operator $\mathrm{Cov}(\phi(x), \phi(y))$. With the kernel matrices $K^X \in \mathbb{R}^{N\times N}$ and $K^Y \in \mathbb{R}^{N\times N}$, the HSIC value is defined as

$$\max_{u,v} = \frac{\mathrm{trace}(\hat{K}^u \hat{K}^v)}{(n-1)^2}$$

By adding a constraint to weight vectors u and v, we obtain

$$\max_{u,v} = \frac{\mathrm{trace}(\hat{K}^u \hat{K}^v)}{(n-1)^2}$$

$$s.t. \|u\|_c \leq s_x,\ \|v\|_c \leq s_y,$$

where c indicates the norm and s_x and s_y are user-defined constants to adjust the degree of sparsity. With (S)CCA-HSIC, the norms of the two views do not need to be the same. Using (S)CCA-HSIC with l_2 norm is equivalent to CCA-HSIC. In this thesis, we examine (S)CCA-HSIC as a dense model with $c = 2$ instead of a sparse model.

2.4.4 Kernel matrix centering

Kernel matrix consists of pairwise inner products of the observations translated into feature space. If the convex hull of the data in feature space is far away from the origin, all inner product values get values similar to each other, which results in the data being ill-conditioned [25]. It is safe to say that in many kernel applications, an ill-conditioned data set often performs worse than a data set which has a hull near the origin. In fact, in many applications of kernel methods, kernel centering is required. For example, CCA-HSIC requires kernel centering.

When the data in a kernel matrix is uncentered, the data can be moved towards the origin in feature space, which is called centering of the kernel matrix. Next, we will show how to center train and test kernel matrices, as described in [26].

Centering train kernel matrix An arbitrary data set can be centered to have a mean of zero ($\frac{1}{n}\sum_{i=1}^{n} x_n = \vec{0}$) through $\tilde{x}_i = x_i - \frac{1}{n}\sum_{j=1}^{n} x_j$. Using a

similar intuition, a data set that has been translated into feature space can be centered in feature space:

$$\tilde{\phi}(x_i) = \phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(x_j) \tag{2.11}$$

However, as it is not often possible nor necessary to explicitly calculate the translated data vectors, we will center the inner product values of the kernel matrix instead. Similar to our definition $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$, we can write the same for centered kernel matrix:

$$\tilde{K}(x_i, x_j) = \langle\tilde{\phi}(x_i), \tilde{\phi}(x_j)\rangle \tag{2.12}$$

By substituting $\tilde{\phi}(x_i)$ and $\tilde{\phi}(x_j)$ in (2.12) with (2.11), we get

$$\begin{aligned}
\tilde{K}(x_i, x_j) &= \langle\phi(x_i) - \frac{1}{n}\sum_{p=1}^{n}\phi(x_p), \phi(x_j) - \frac{1}{n}\sum_{q=1}^{n}\phi(x_q)\rangle \\
&= (\phi(x_i)^\intercal - \frac{1}{n}\sum_{p=1}^{n}\phi(x_p)^\intercal)(\phi(x_j) - \frac{1}{n}\sum_{q=1}^{n}\phi(x_q)) \\
&= \phi(x_i)^\intercal\phi(x_j) - \frac{1}{n}\sum_{p=1}^{n}\phi(x_p)^\intercal\phi(x_j) \\
&\quad - \frac{1}{n}\sum_{q=1}^{n}\phi(x_i)^\intercal\phi(x_q) + \frac{1}{n^2}\sum_{p=1}^{n}\sum_{q=1}^{n}\phi(x_p)^\intercal\phi(x_q)
\end{aligned}$$

Since $\phi(x_i)^\intercal\phi(x_j) = K(x_i, x_j)$, the previous can be written as

$$\begin{aligned}
\tilde{K}(x_i, x_j) = K(x_i, x_j) &- \frac{1}{n}\sum_{p=1}^{n}K(x_p, x_j) - \frac{1}{n}\sum_{q=1}^{n}K(x_i, x_q) \\
&+ \frac{1}{n^2}\sum_{p=1}^{n}\sum_{q=1}^{n}K(x_p, x_q)
\end{aligned}$$

This can be written in matrix form,

$$\tilde{K} = K - K1_n - 1_nK + 1_nK1_n,$$

where 1_n is a $n \times n$ matrix consisting of all $\frac{1}{n}$'s.

Centering test kernel matrix Non-centered test kernel is defined similarly to train kernel, except that the inner product is calculated between the test and train set observations,

$$K_{ij}^{test} = \langle \phi(x_i^{test}), \phi(x_j^{train}) \rangle,$$

Then, the test kernel can be centered similarly to the centering of train kernel

$$\tilde{K}_{ij}^{test} = \langle \tilde{\phi}(x_i^{test}), \tilde{\phi}(x_j^{train}) \rangle,$$

which can be reformulated as

$$\begin{aligned}
\tilde{K}^{test}(x_i, x_j) &= \langle \tilde{\phi}(x_i^{test}), \tilde{\phi}(x_j^{train}) \rangle \\
&= (\phi(x_i^{test})^\intercal - \frac{1}{n}\sum_{p=1}^{n} \phi(x_p^{train})^\intercal)(\phi(x_j^{train}) - \frac{1}{n}\sum_{q=1}^{n} \phi(x_q^{train})) \\
&= \phi(x_i^{test})^\intercal \phi(x_j^{train}) - \frac{1}{n}\sum_{p=1}^{n} \phi(x_p^{train})^\intercal \phi(x_j^{train}) \\
&\quad - \frac{1}{n}\sum_{q=1}^{n} \phi(x_i^{test})^\intercal \phi(x_q^{train}) + \frac{1}{n^2}\sum_{p=1}^{n}\sum_{q=1}^{n} \phi(x_p^{train})^\intercal \phi(x_q^{train}) \\
&= K(x_i^{test}, x_j^{train}) - \frac{1}{n}\sum_{p=1}^{n} K(x_p^{train}, x_j^{train}) \\
&\quad - \frac{1}{n}\sum_{q=1}^{n} K(x_i^{test}, x_q^{train}) + \frac{1}{n^2}\sum_{p=1}^{n}\sum_{q=1}^{n} K(x_p^{train}, x_q^{train}),
\end{aligned}$$

where n is the number of observation in the train set. This can then be written in matrix form,

$$\tilde{K}^{test} = K^{test} - 1_{n_{test}} K^{train} - K^{test} 1_{n_{train}} + 1_{n_{test}} K^{train} 1_{n_{train}},$$

where $1_{n_{test}}$ is a $n_{test} \times n_{train}$ matrix with all values $\frac{1}{n^{test}}$, and $1_{n_{train}}$ is a $n_{train} \times n_{train}$ matrix with all values $\frac{1}{n_{train}}$.

2.5 The state of the research in kernelized CCA variants

As mentioned earlier in this thesis, extending CCA to finding non-linear relationships using kernel methods was first proposed in Lai and Fyfe (2000).

The research continues in the late 2010s, with a focus on scalability, robustness to irrelevant variables, interpretability and the capability to extract complex, non-linear relationships. Many of the recent focal points of research are shortcomings of the first kernelized CCA method, the KCCA. In this chapter, we will go through research by Akaho (2001) [1], Chang et al. (2013) [2], Lopez-Paz et al. (2014) [27], Wang & Livescu (2016) [28], Yoshida et al. (2017) [29], Uurtio et al (2018, 2019) [3] [6].

One of the earliest works regarding kernelized CCA models was by Akaho in 2001. They ran KCCA in two experiments with simulated data. The first experiment contained a multivariate setting with two trigonometric relationships. KCCA with a Gaussian kernel and regular CCA were executed on the data, set to extract two relationships. As the result, KCCA found the relationships with higher correlation coefficients than regular CCA. The second experiment had random-generated class points, around which random points were generated. Also in the second experiment, KCCA showed capability for recognizing non-linear relationships while regular CCA did not. In the research, they found that choosing the right regularisation parameter is important, and suggested using methods like cross-validation, resampling methods or empirical Bayesian approaches for determining the regularisation parameter. They also concluded that kernel type selection is crucial for the performance of KCCA [1].

After the development of Kernelized CCA, three main problems have arisen. First, the canonical weights of KCCA are not easily interpretable, therefore drawing conclusions about the relationships can be difficult. Second, KCCA is a dense method and as such is not robust against large number of noise variables. Third, calculating kernel matrices in KCCA can become computationally intensive. To address these, Chang et al. proposed CCA-HSIC, which is a kernelized CCA variant that uses Hilbert-Schmidt independence criterion and the centered kernel target alignment. In their proposition, they experimented with simulated data, which contained trigonometric relationships and a linear relationship. They ran CCA-HSIC and other CCA variants, such as KCCA and regular CCA and found out that CCA-HSIC and KCCA managed to extract the non-linear relationships, whereas CCA could not. Although KCCA managed to retrieve all relationships, it did not retrieve the relationships as precisely as CCA-HSIC. For KCCA, it took five components in total to find all the relationships, indicating lack of robustness against irrelevant variables [2].

Another attempt to address the issue of computational intensiveness of KCCA was by Lopez-Paz et al. in 2014 [27]. They proposed randomized non-linear CCA (RCCA), which approximates kernel matrices by random Fourier features [30]. However, RCCA approximates the kernel canonical

correlation of KCCA and, despite improving the scalability, it does not solve the problem of interpretability.

A branch from RCCA was kernel non-linear orthogonal iterations or KNOI, which was proposed by Wang & Livescu (2016) [28] to address the issue of high memory requirements of RCCA. KNOI uses a memory efficient stochastic optimization algorithm to approximate KCCA. In their research, KNOI performed better than RCCA in terms of speed and test correlation.

Two-stage kernel CCA (TSKCCA) is a kernelized CCA variant proposition by Yoshida et al. in 2017 [29]. The method is based on the framework of multiple kernel learning. It aims to tackle the shortcomings of KCCA, lack of feature selection and difficulties in capturing multiple canonical components. In their proposition, they experimented with the method in multiple different settings. TSKCCA managed to extract the correct relationships between the data sets in cases of single and multivariate, non-linear and sparse data sets.

(S)CCA-HSIC was proposed in 2018 by Uurtio et al. [3] for finding sparse non-linear relationships from two-view data as a method that applies Hilbert-Schmidt independence criterion. They compared the performance of (S)CCA-HSIC to other CCA variants on simulated and two real-world data sets. In the simulated experiments, they compared the methods in terms of robustness to noise variables, ability to extract relationships as the related variables increase, scalability to large number of variables and ability to extract complicated relationships. In the simulated experiments, they defined linear, sinusoidal and hyperbolic multivariate relationships. With simulated data, (S)CCA-HSIC performs better or equally good to the second best method in each category. Also with the real-world data, (S)CCA-HSIC showcases capability to discover relationships.

In 2019, Uurtio et al. proposed GradKCCA as a kernelized CCA variant that does not require the use of kernel matrices [6]. It is designed to be a faster alternative to KCCA and CCA-HSIC. It can also be used with l_1 norm, which makes it sparse. In their proposition, they compared gradKCCA to other CCA variants on a data set that has monotonic and multivariate trigonometric relationships. In their setting, a varying number of noise variables was used. In the monotonic relationship experiment, robustness to noise variables of gradKCCA was better than that of DCCA, KNOI and (S)CCA-HSIC. Only KCCA performed nearly at the same level as gradKCCA. In the second experiment, trigonometric relationships were created and also the number of variables was increased. There, in terms of correlation over test set, gradKCCA performed equally well to the second best performing variant, (S)CCA-HSIC. However, gradKCCA topped (S)CCA-HSIC in terms of F1 score. In the third experiment, the sample size was varied and a data set containing monotonic and trigonometric relationships was generated. Among

the compared methods, GradKCCA was the fastest with both relationship types. However, in terms of correlation over test set, gradKCCA did not stand out.

Chapter 3

Research material and methods

In this thesis, we run four CCA variants on simulated data and real-world data sets. In this chapter, we go through the details of how the experiments are run. First, we explain the settings of the experiments with simulated data. Then, we explain how the experiments with real-world data sets are executed.

All experiments in this thesis are implemented using MATLAB R2017a and computed on a Macbook Pro Mid 2012 (2.6 GHz Intel Core i7, 8 GB 1600 MHz DDR3).

3.1 Simulations

We run 12 different experiments on computer-generated data sets. In all experiments, we have two views with a low number of variables in each view. In each experiment, we vary the number and the type of relationships created between the variables of the views. Then, we examine how CCA, KCCA, gradKCCA and (S)CCA-HSIC perform in each setup.

In each experiment, all four CCA variants are run 10 times. For each run, F1 score and correlation over train and test sets are calculated and the canonical weight vectors u and v are stored. Then, mean values are calculated for each measure and weight vector.

In the simulated experiments, the data set has $N = 500$ observations and all variables in the data are standardized to have a zero mean and unit variance. The data is then divided into train and test sets. The ratio of train and test sets is set to $2 : 1$ in all experiments. Train set is used for calculating the weight vectors and calculation of correlation over train set. Test set is used for estimation of how well the results of CCA variants generalise for new data from the same distribution.

F1 score is calculated based on ground truth vectors, which we define in the beginning of each simulation. F1 score is calculated as $F1 = \frac{2TP}{2TP+FP+FN}$, where TP is number of true positives, FP is the number of false positives and FN is the number of false negatives. For F1 score, we transform the elements of the weights to have binary values so that if the absolute value of an element is larger than 0.05, then the element is 1, otherwise it is 0. Then, we compare each element of vectors u and v to the corresponding ground truth vectors, which are also binary, and determine the number of TP, FP and FN. The explicit projections for KCCA are not available, therefore we calculate approximations of the weight vectors in input space by $\tilde{u} = X_{train}^{\intercal}\alpha$ and $\tilde{v} = Y_{train}^{\intercal}\beta$, as suggested in [3].

Another metric used for measuring the performance of the methods in simulated settings is R precision. It is defined as $\frac{r}{R}$, where r is the number of relevant items among R retrieved items. It is applied so that we determine r by how many of the R highest coefficients in the retrieved canonical weights are marked ones in the ground truth vectors, where R is the total number of relevant variables in the ground truth.

We solve CCA through standard eigenvalue problem. For KCCA, we use linear and polynomial kernels depending on the experiment. We center the test and train kernels and solve KCCA as a generalised eigenvalue problem. Regularisation constant is determined using k-fold cross-validation with 5 folds, as demonstrated in [31]. GradKCCA uses l_2 norm and norm constants 1.0 for both views. GradKCCA uses polynomial or Gaussian kernels depending on the experiment. The gradKCCA formula is solved using an alternating projected gradient approach proposed in [6], which is adapted with permission into Algorithm 1. Repetitions are limited to 100 and stopping criterion will be set to 10^{-10}. For (S)CCA-HSIC, we only use Gaussian kernel, since HSIC requires a universal kernel. (S)CCA-HSIC is solved using a projected stochastic gradient ascent algorithm, as proposed in [3], with 5 random initializations. The algorithm presented in [3] is adapted with permission into Algorithm 2. For all polynomial kernels we use degree of 2. For all Gaussian kernels, bandwidth parameter σ is chosen using the "median trick". The noise variables used in simulations are defined as follows: $\xi_1 \sim N(0, 0.05^2)$, $\xi_2 \sim N(0, 0.1^2)$ and $\xi_3 \sim N(0, 0.15^2)$.

3.2 Real-world data

In real-world experiments we run CCA, KCCA, gradKCCA and (S)CCA-HSIC on two real-world data sets, retrieving three relationships. We compare the results and investigate what kind of relationships each variant finds. Each

Algorithm 1 gradKCCA alternating projected gradient

Input
  X, Y matrices of measurements
  M number of components
  R repetitions
  δ convergence limit
  P_x, P_y norms of u and v
  s_x, s_y l_1 or l_2 norm constraints for u and v
  d_x, d_y hyperparameters for k^x and k^y
Output
  U, V weight vectors
for all $m = \{1, 2, ..., M\}$ **do**
  for all $r = \{1, 2, ..., R\}$ **do**
    Initialize $u_m r$ and $v_m r$
    Compute $k^x(u), k^y(v)$
    repeat
      Compute $\rho_{old} = \rho(u, v)$
      Compute $\nabla\rho_u = \frac{\partial\rho(u,v)}{\partial u}$
      Update $u_{mr} = \prod_{\|.\|_{P_x} \leq s_x} (u_{mr} + \gamma\nabla u)$
      Re-compute $k^x(u)$
      Compute $\nabla\rho_v = \frac{\partial\rho(u,v)}{\partial v}$
      Update $v_{mr} = \prod_{\|.\|_{P_x} \leq s_x} (v_{mr} + \gamma\nabla v)$
      Re-compute $k^y(v)$
      Compute $\rho_{current} = \rho(u, v)$
    until $|\rho_{old} - \rho_{current}|/|\rho_{old} + \rho_{current}| < \delta$
    $\rho_r = \rho_{current}$, $u_r = u_{mr}$, $v_r = v_{mr}$
  Select $r^* = \arg\max_r \rho_r$
  Store $U(:, m) = u_{r*}$, $V(:, m) = v_{r*}$
  Deflate $X^{(m)}, Y^{(m)}$ by $U(:, m)$ and $V(:, m)$
return U, V

Algorithm 2 CCA-HSIC projected stochastic gradient ascent

Input
  X, Y matrices of measurements
  M number of components
  R repetitions
  δ convergence limit
  P_x, P_y norms of u and v
  s_x, s_y l_1 or l_2 norm constraints for u and v
  σ_u, σ_v standard deviations of the Gaussian kernels
Output
  U, V weight vectors
for all $m = \{1, 2, ..., M\}$ **do**
  for all $r = \{1, 2, ..., R\}$ **do**
    Initialize u_{mr} and v_{mr}
    *Compute K^u, K^v, $\tilde{K}^u$ and $\hat{K}^v$
    repeat
      *Compute $f_{old} = \rho(u, v)$
      Compute $\nabla u = \frac{\partial \rho(u,v)}{\partial u}$
      Update $u_{mr} = \prod_{\|\cdot\|_{P_x} \leq s_x} (u_{mr} + \gamma \nabla u)$
      *Compute K^u and $\tilde{K}^u$
      Compute $\nabla v = \frac{\partial \rho(u,v)}{\partial v}$
      Update $v_{mr} = \prod_{\|\cdot\|_{P_x} \leq s_x} (v_{mr} + \gamma \nabla v)$
      *Compute K^v and $\tilde{K}^v$
      Compute $f_{current} = \rho(u, v)$
    until $|f_{old} - f_{current}|/|f_{old} + f_{current}| < \delta$
    $f_r = f_{current}$, $u_r = u_{mr}$, $v_r = v_{mr}$
  Select $r^* = \arg\max_r f_r$
  Store $U(:, m) = u_{r*}$, $V(:, m) = v_{r*}$
  Deflate $X^{(m)}, Y^{(m)}$ by $U(:, m)$ and $V(:, m)$
return U, V

method is run ten times on both data sets, with a repeatedly randomized division into training and test sets. The ratio of training and test sets is 2:1. On each run, we collect weights u and v, then select the run which produces the highest average correlation over the test set. Then, we analyze the selected run.

The two real-life data sets are the body fat data set and the Boston housing data set. The body fat data set consists of 252 observations of body fat related variables of American men. The Boston housing data set has 506 observations of housing deals in the Boston area. The Boston housing data set is known to contain non-linear relationships [2].

For real-life experiment, we solve CCA through standard eigenvalue problem. For KCCA, we use linear or polynomial kernels depending on the experiment. We center the test and train kernels and solve KCCA as a generalised eigenvalue problem. Regularisation constant is set to $c = 0.02$ for both views. GradKCCA uses l_2 norm and norm constraints 1.0 for both views. GradKCCA is run twice, once using polynomial kernels and once using Gaussian kernels on both data sets. The gradKCCA formula is solved using an alternating projected gradient approach, which is shown in Algorithm 2 [6]. Repetitions are limited to 100 and stopping criterion is set to 10^{-10}. For (S)CCA-HSIC, we only use a Gaussian kernel, since HSIC requires a universal kernel. (S)CCA-HSIC is solved using a projected stochastic gradient ascent algorithm as proposed in [3] and shown in Algorithm 2, with five random initializations.

Chapter 4

Results

4.1 Experiments with simulated data

4.1.1 Monotonic relationships

Experiment 1. In the following experiment, we evaluate the performances of the CCA variants to detect linear combinations with data sets defined as follows:

$$y_1 = x_1 + x_2 + x_3 + \xi_1$$

$$y_2 = x_4 + x_5 + x_6 + \xi_2$$

$$y_3 = x_7 + x_8 + x_9 + \xi_3,$$

where x_1 to x_9 and y_1 to y_3 are variables of the views $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times q}$, respectively ($p = 10$, $q = 8$). The corresponding ground truth vectors are $u_{g1} = [1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, $v_{g1} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, $u_{g2} = [0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0]$, $v_{g2} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$ and $u_{g3} = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0]$, $v_{g3} = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$.

As can be seen in Figure 4.1, CCA and KCCA (linear kernel) perform the best with perfect precision and F1 of 0.82 and 0.79. Both gradKCCA (linear kernel) and (S)CCA-HSIC put the highest weights on the right variables, but do not manage to separate the relationships. Therefore, the F1 scores are 0.52 for gradKCCA (linear kernel) and 0.53 for (S)CCA.

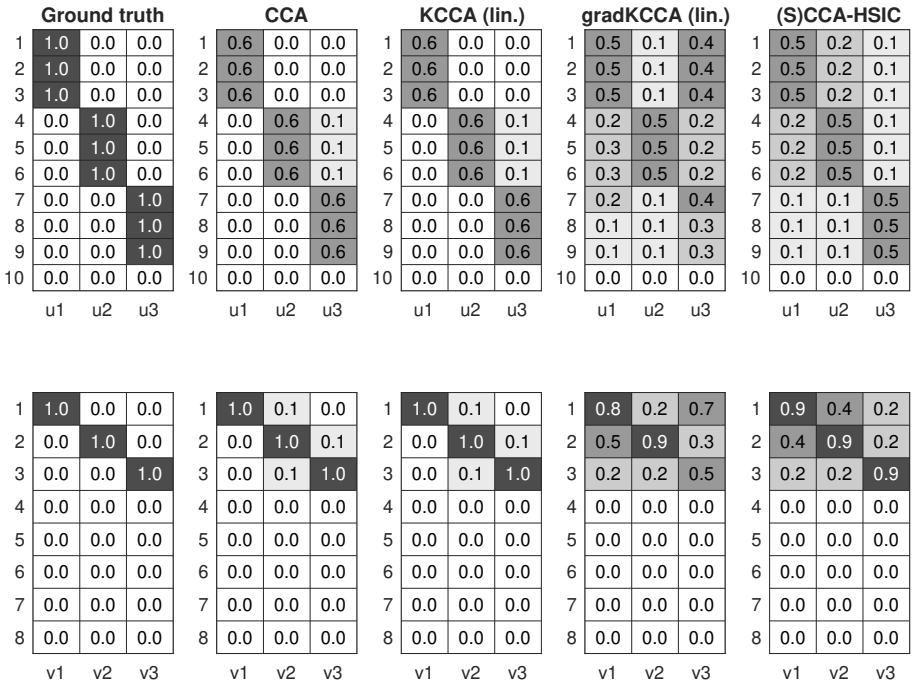
Experiment 2. In this experiment, we create two multivariate relationships between views X and Y as follows:

$$y_1 + y_2 = x_1 + x_2 + \xi_1$$

$$y_3 + y_4 = x_3 + x_4 + \xi_2,$$

	F1	R precision	ρ_{tr1}	ρ_{tr2}	ρ_{tr3}	ρ_{tt1}	ρ_{tt2}	ρ_{tt3}
CCA	0.79 (0.08)	1.00 (0.00)	1.00 (0.00)	0.98 (0.00)	0.96 (0.00)	0.99 (0.00)	0.98 (0.00)	0.96 (0.01)
KCCA (lin.)	0.82 (0.09)	1.00 (0.00)	1.00 (0.00)	0.98 (0.00)	0.96 (0.00)	0.99 (0.00)	0.98 (0.00)	0.96 (0.01)
gradKCCA (lin.)	0.52 (0.03)	0.73 (0.08)	0.99 (0.00)	0.92 (0.07)	0.94 (0.04)	0.97 (0.01)	0.90 (0.08)	0.94 (0.04)
(S)CCA-HSIC	0.53 (0.04)	0.90 (0.12)	0.99 (0.00)	0.97 (0.01)	0.96 (0.01)	0.99 (0.00)	0.97 (0.01)	0.96 (0.01)

(a) F1 score and correlations over train and test sets for CCA variants. Standard deviation is marked in parenthesis.

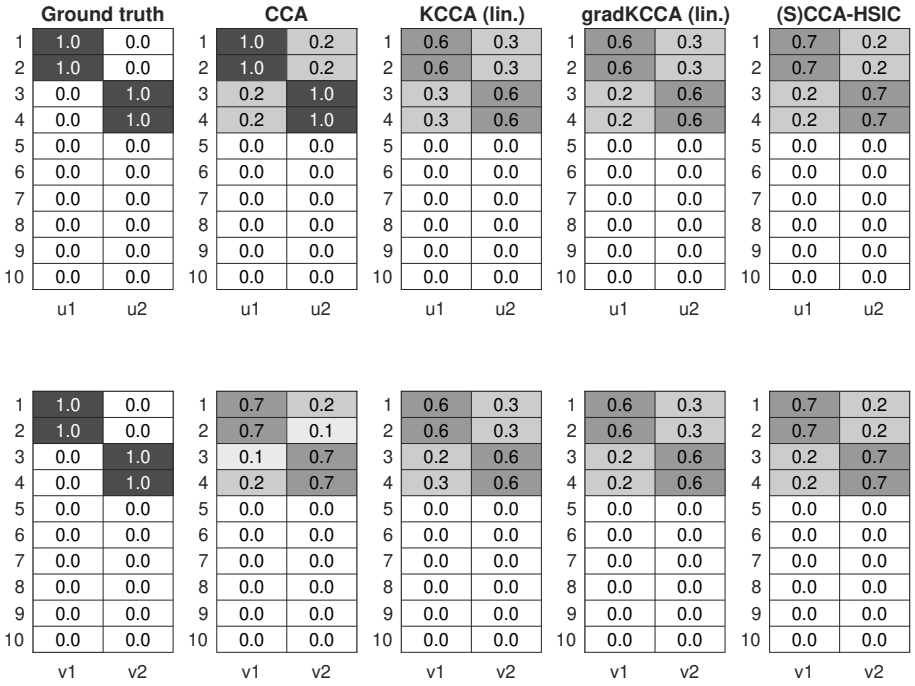


(b) Mean canonical weights over 10 runs of first, second and third components.

Figure 4.1: The results of CCA variants on a data set with three linear relationships.

	F1	R precision	ρ_{tr1}	ρ_{tr2}	ρ_{tt1}	ρ_{tt2}
CCA	0.71 (0.06)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
KCCA (lin.)	0.69 (0.06)	0.94 (0.14)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
gradKCCA (lin.)	0.65 (0.05)	0.89 (0.18)	0.99 (0.00)	0.96 (0.04)	0.99 (0.00)	0.96 (0.05)
(S)CCA-HSIC	0.68 (0.11)	1.00 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)

(a) F1 score and correlations over train and test sets for CCA variants. Standard deviation is marked in parenthesis.



(b) Mean canonical weights over 10 runs of first, second and third components.

Figure 4.2: The results of CCA variants on a data set with two linear multivariate relationships.

where x_1, x_2, y_1, and y_2 represent the variables of views $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times p}$, respectively ($p = 6$, $q = 6$). In this case, the ground truth vectors are $u_{g1} = [1\ 1\ 0\ 0\ 0\ 0]$, $v_{g1} = [1\ 1\ 0\ 0\ 0\ 0]$, and $u_{g2} = [0\ 0\ 1\ 1\ 0\ 0]$, $v_{g2} = [0\ 0\ 1\ 1\ 0\ 0]$.

As Figure 4.2 shows, CCA has the highest F1 score 0.71 and the R precision is 1.0. CCA does not manage to separate the relationships in this experiment as well as in Experiment 1. KCCA (linear kernel) has an F1 score of 0.69, gradKCCA (linear kernel) scores 0.65, and (S)CCA-HSIC 0.68. None of the kernelized methods manage to separate the relationships, either.

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	0.95 (0.08)	1.00 (0.00)	0.92 (0.01)	0.91 (0.01)	[1 0 0 0 0 0]	[1 0 0 0 0 0]
KCCA (quad.)	0.48 (0.51)	0.50 (0.53)	0.91 (0.00)	0.88 (0.01)	[0.5 0.2 0.2 0.1 0.3 0.1]	[0.5 0.2 0.2 0.2 0.2 0.2]
gradKCCA (quad.)	0.70 (0.15)	1.00 (0.00)	0.90 (0.00)	0.89 (0.01)	[1 0 0 0 0 0]	[1 0 0.1 0.1 0 0]
gradKCCA (Gauss.)	0.51 (0.08)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	[1 0 0.1 0 0.1 0]	[1 0.1 0 0.1 0 0.1]
(S)CCA-HSIC	0.88 (0.11)	1.00 (0.00)	0.87 (0.01)	0.87 (0.02)	[1 0 0 0 0 0]	[1 0 0 0 0 0]

Figure 4.3: F1 score, correlations over train and test sets and canonical weights of CCA variants on a data set with a cubic relationship. Standard deviation is marked in parenthesis.

Experiment 3. Next, we experiment how CCA variants find a cubic relationship, which is a monotonic, non-linear relationship. We define

$$y_1 = x_1^3 + \xi_1,$$

where x_1 and y_1 are the first variables of view $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times p}$ ($p = 6$, $q = 6$). The ground truth vectors are $u_{g1} = [1\ 0\ 0\ 0\ 0\ 0]$, $v_{g1} = [1\ 0\ 0\ 0\ 0\ 0]$.

Figure 4.3 shows that CCA is able to discover a cubic relationship with the highest F1 score 0.95. It is followed by (S)CCA-HSIC with F1 score 0.88. GradKCCA (quadratic kernel) has an F1 score of 0.70, KCCA (quadratic kernel) scores 0.48, and gradKCCA (Gaussian kernel) scores 0.51. In terms of R precision, all methods score 1.0 except KCCA (quadratic kernel), which scores 0.50.

4.1.2 Non-monotonic relationships

Experiment 4. In the following experiment, we explore how the compared CCA variants manage to find a single quadratic relationship between views X and Y. The relationship is defined as follows:

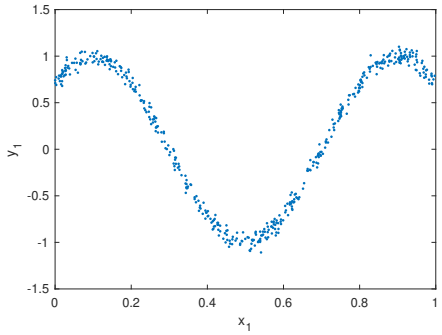
$$y_1 = x_1^2 + \xi_1,$$

where x_1 and y_1 are the first variables of views $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times p}$, respectively ($p = 6$, $q = 6$). The ground truth vectors are $u_g = [1\ 0\ 0\ 0\ 0\ 0]$, $v_g = [1\ 0\ 0\ 0\ 0\ 0]$.

As Figure 4.4 shows, CCA is expectedly unable to discover the quadratic relationship having F1 value 0.29. KCCA (quadratic kernel) also has a low F1 value (0.38), but the canonical weights are slightly closer to the ground truth vectors compared to CCA. GradKCCA (Gaussian kernel) has an F1 score 0.32, but performs worse relative to gradKCCA with a quadratic kernel (0.57). The best performance is given by (S)CCA-HSIC, which has an F1 score of 0.77 and has weights very close to the ground truth vectors. It is also the only method with an R precision 1.0.

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	0.29 (0.06)	0.25 (0.35)	0.24 (0.03)	0.02 (0.02)	[0.5 0.5 0.5 0.5 0.5 0.5]	[0.7 0.5 0.4 0.5 0.3 0.6]
KCCA (quad.)	0.38 (0.28)	0.70 (0.48)	0.62 (0.04)	0.45 (0.07)	[0.7 0.1 0.1 0.2 0.2 0.1]	[0.7 0.1 0.1 0.2 0.2 0.2]
gradKCCA (quad.)	0.57 (0.14)	0.90 (0.32)	0.76 (0.05)	0.73 (0.11)	[0.9 0.1 0 0 0.1]	[0.9 0.1 0 0.1 0 0]
gradKCCA (Gauss.)	0.32 (0.08)	0.75 (0.35)	0.99 (0.00)	0.99 (0.00)	[0.7 0.1 0.2 0.1 0.2 0.2]	[0.7 0.2 0.2 0.3 0.1 0.2]
(S)CCA-HSIC	0.77 (0.23)	1.00 (0.00)	0.77 (0.02)	0.76 (0.02)	[1 0 0 0 0 0]	[1 0 0 0 0 0]

Figure 4.4: F1 score, correlations over train and test sets and canonical weights of CCA variants on a data set with a single quadratic relationship. Standard deviation is marked in parenthesis.



(a) The plot of variable y_1 as a function of x_1 after changing the standard deviation to 1 and mean to 0.

	F1	R precision	ρ_{tr}	ρ_{tt}	$\tilde{u}$	$\tilde{v}$
CCA	0.31 (0.06)	0.25 (0.26)	0.24 (0.03)	0.02 (0.03)	[0.5 0.6 0.5 0.6 0.5 0.4]	[0.6 0.5 0.4 0.6 0.3 0.6]
KCCA (quad.)	0.29 (0.07)	0.30 (0.26)	0.49 (0.03)	0.12 (0.10)	[0.6 0.3 0.2 0.3 0.2 0.2]	[0.1 0.2 0.4 0.3 0.5 0.3]
gradKCCA (quad.)	0.29 (0.15)	0.30 (0.26)	0.64 (0.03)	0.55 (0.06)	[0.1 0.3 0.4 0.3 0.2 0.2]	[0.6 0.3 0.1 0.1 0.1 0.2]
gradKCCA (Gauss.)	0.43 (0.09)	1.00 (0.00)	0.99 (0.00)	0.99 (0.00)	[1 0 0.1 0.1 0 0.1]	[1 0.1 0.1 0.1 0.1 0.1]
(S)CCA-HSIC	0.83 (0.16)	1.00 (0.00)	0.04 (0.03)	0.06 (0.04)	[1 0 0 0 0 0]	[1 0 0 0 0 0]

(b) F1 score, correlations over train and test sets and canonical weights of CCA variants. Standard deviation is marked in parenthesis.

Figure 4.5: The results of CCA variants on a trigonometic relationship.

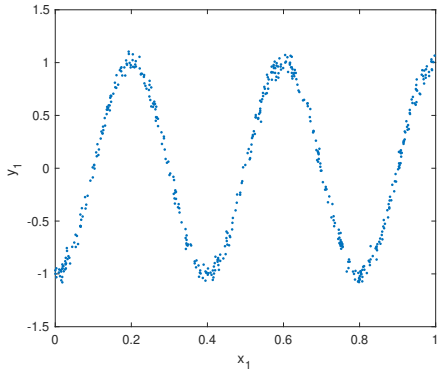
Experiment 5. In the next experiment, we create a single trigonometric relationship between views $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^{N \times p}$ ($p = 6$, $q = 6$). The relationship is part of a sine wave and is defined as

$$y_1 = \sin(x_1) + \xi_1,$$

where $x_1 \sim U(\frac{\pi}{4}, \frac{2.5\pi}{4})$ and y_1 are variables of views X and Y, respectively.

Figure 4.5a is the plot of variable y_1 as a function of x_1. There we can see that the relationship is non-monotonic, in that it has two local maxima and three local minima. It is rather similar in shape to the parabola-shaped relationship in Experiment 4, where gradKCCA with a quadratic kernel and (S)CCA-HSIC performed well.

Figure 4.5b shows that, unlike in Experiment 1, gradKCCA with a



(a) The plot of variable y_1 as a function of x_1 after changing the standard deviation to 1 and mean to 0.

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	0.25 (0.07)	0.20 (0.35)	0.24 (0.03)	0.03 (0.02)	[0.5 0.4 0.5 0.5 0.4 0.6]	[0.5 0.5 0.4 0.6 0.4 0.7]
KCCA (quad.)	0.31 (0.02)	0.25 (0.35)	0.47 (0.04)	0.07 (0.05)	[0.4 0.3 0.4 0.3 0.3 0.2]	[0.3 0.3 0.3 0.3 0.4 0.3]
gradKCCA (quad.)	0.36 (0.17)	0.45 (0.37)	0.64 (0.01)	0.55 (0.06)	[0.3 0.2 0.4 0.1 0.2 0.2]	[0.6 0.1 0.1 0.3 0.2 0.1]
gradKCCA (Gauss.)	0.26 (0.09)	0.10 (0.21)	0.99 (0.00)	0.99 (0.00)	[0.3 0.4 0.3 0.2 0.3 0.2]	[0.2 0.4 0.4 0.3 0.3 0.4]
(S)CCA-HSIC	0.27 (0.08)	0.15 (0.34)	0.13 (0.05)	0.05 (0.04)	[0.4 0.3 0.4 0.2 0.3 0.3]	[0.2 0.4 0.4 0.4 0.3 0.3]

(b) F1 score, correlations over train and test sets and canonical weights of CCA variants. Standard deviation is marked in parenthesis.

Figure 4.6: The results of CCA variants on a long trigonometric relationship.

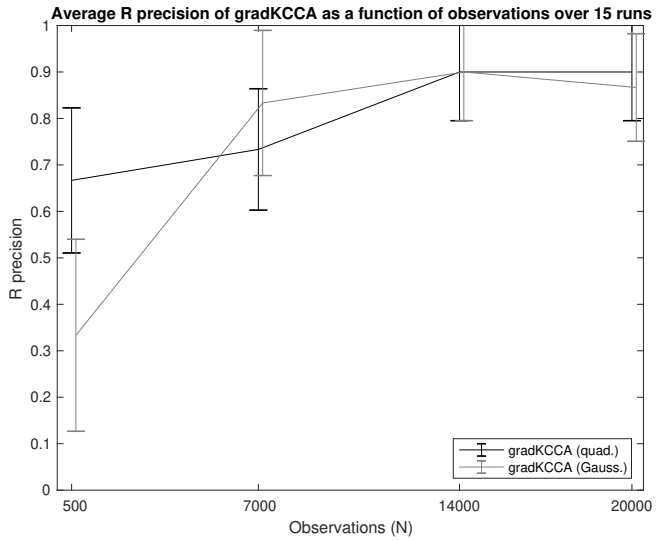
quadratic kernel no longer yields canonical weights accurately, instead it has a low F1 score (0.29). KCCA (quadratic kernel) scores 0.30 and is also unable to discover the relationship. GradKCCA (Gaussian kernel), however, retrieves the relationship with a score 0.43 and a perfect R precision. (S)CCA-HSIC performs the best with the highest F1 score 0.83 and an R precision 1.0.

Experiment 6.1 Next, we experiment how the compared CCA variants perform when the trigonometric relationship between views X and Y is more complex. We define

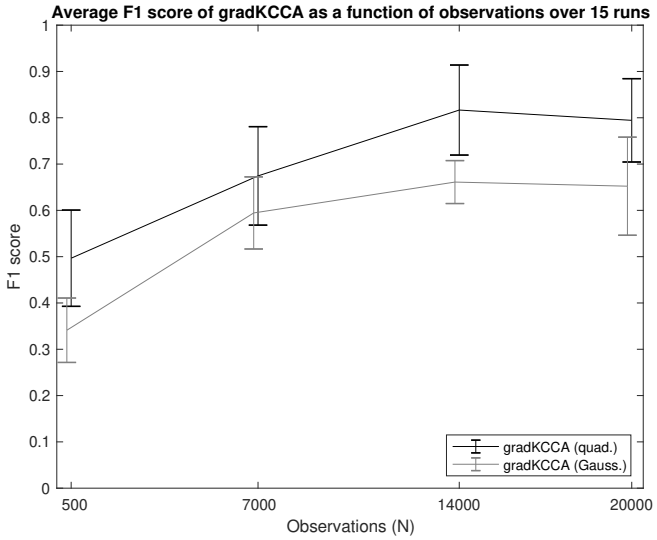
$$y_1 = \sin(x_1) + \xi_1,$$

where $x_1 \sim U(\pi, 5\pi)$ and y_1 are the first variables of views X and Y, respectively. Ground truth vectors are $u_g = [1\ 0\ 0\ 0\ 0\ 0]$, $v_g = [1\ 0\ 0\ 0\ 0\ 0]$. Plot of the relationship in Figure 4.6a shows how the relationship is non-monotonic, having three local maxima and minima.

It can be seen from Figure 4.6b that none of the methods is capable of retrieving the correct weight vectors. The highest F1 score is by gradKCCA quadratic (0.36) and also it has the highest R precision 0.45. However, the retrieved canonical weights are far from the ground truth values. Although



(a) The plot of R precision of gradKCCA with polynomial and Gaussian kernels on a long trigonometric relationship using 95% confidence interval.



(b) The plot of F1 of gradKCCA with polynomial and Gaussian kernels on a long trigonometric relationship using 95% confidence interval.

Figure 4.7: The results of gradKCCA on a long trigonometric relationship when the number of observations is increased.

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	0.27 (0.05)	0.15 (0.24)	0.24 (0.03)	0.02 (0.02)	[0.4 0.6 0.4 0.5 0.4 0.5]	[0.3 0.5 0.5 0.6 0.4 0.6]
KCCA (quad.)	0.50 (0.00)	0.50 (0.00)	0.96 (0.01)	0.95 (0.02)	[0.5 0.2 0.3 0.2 0.2 0.1]	[0.5 0.1 0.2 0.2 0.2 0.2]
gradKCCA (quad.)	0.30 (0.13)	0.45 (0.16)	0.70 (0.03)	0.62 (0.06)	[0.1 0.2 0.5 0.2 0.3 0.1]	[0.9 0.1 0.1 0.1 0.1 0.1]
gradKCCA (Gauss.)	0.26 (0.11)	0.15 (0.24)	0.99 (0.00)	0.99 (0.00)	[0.4 0.4 0.1 0.1 0.3 0.3]	[0.2 0.3 0.3 0.5 0.2 0.4]
(S)CCA-HSIC	0.29 (0.05)	0.20 (0.26)	0.15 (0.06)	0.04 (0.03)	[0.4 0.4 0.4 0.2 0.3 0.3]	[0.3 0.5 0.3 0.3 0.3 0.3]

Figure 4.8: F1 score, correlations over train and test sets and weight vectors for CCA variants on a data set with a circle shaped relationship. Standard deviation is marked in parenthesis.

(S)CCA-HSIC and gradKCCA (Gaussian kernel) were able to discover the relationship in Experiment 5, here neither of the methods could find the relationship.

Experiment 6.2 To investigate the situation further, we set up another experiment. We define a data set with the same relationship, and run the experiment with $N = 500, 7000, 14000, 20000$ observations and variable counts $p = 3$, $q = 3$. Due to the high computational intensity of KCCA and (S)CCA-HSIC, we only inspect gradKCCA with quadratic and Gaussian kernels.

Figure 4.7 shows that with 500 observations, the F1 score is 0.49 (quadratic kernel) and 0.34 (Gaussian kernel). As the number of observations increase to 14000, the F1 score of gradKCCA with a quadratic kernel increases to 0.74 and gradKCCA with a Gaussian kernel increases to 0.61. In this experiment, there are fewer noise variables compared to the original experiment. The original experiment had $p = 6$ and $q = 6$ variables compared to the $p = 3$ and $p = 3$ variables used in this experiment. For 500 observations, this experiment obtained F1 score 0.49, whereas with the original setting F1 score 0.36 was obtained.

Experiment 7. Next, we define a relationship between first variables of views X and Y so that the relation is shaped like a circle.

$$y_1 = \pm\sqrt{1 - x_1^2} + \xi_1,$$

where x_1 and y_1 are the first variables of views X and Y, respectively. Here, plus and minus sign means that half of the samples have a plus sign and the other half a minus sign. Ground truth vectors are $u_g = [1\ 0\ 0\ 0\ 0\ 0]$, $v_g = [1\ 0\ 0\ 0\ 0\ 0]$.

Figure 4.8 shows that KCCA (quadratic kernel) has the highest F1 score and R precision (both 0.50). However, the weight vector has a lot of weight on incorrect variables. The F1 scores of (S)CCA-HSIC, gradKCCA and CCA

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	0.50 (0.07)	0.55 (0.11)	0.23 (0.03)	0.06 (0.04)	[0.1 0.1 0.1 0.1 0.1 0.1]	[0.9 1 0.1 0.1 0.1 0.1]
KCCA (quad.)	0.56 (0.10)	0.63 (0.18)	0.98 (0.02)	0.97 (0.02)	[0.7 0.1 0.1 0.1 0.1 0]	[0.5 0.6 0.3 0.2 0.2 0.3]
gradKCCA (quad.)	0.77 (0.19)	0.90 (0.21)	0.71 (0.04)	0.69 (0.05)	[0.3 0.8 0.1 0 0 0]	[0.6 0.7 0.1 0 0.1 0.1]
gradKCCA (Gauss.)	0.66 (0.09)	0.90 (0.13)	1.00 (0.00)	1.00 (0.00)	[0.6 0.5 0.1 0 0.1 0]	[0.7 0.7 0.1 0 0.1 0.1]
(S)CCA-HSIC	0.63 (0.15)	0.70 (0.26)	0.05 (0.04)	0.06 (0.05)	[0.9 0.1 0.1 0.1 0.1 0.1]	[0.6 0.6 0.1 0.1 0.1 0.1]

Figure 4.9: F1 score, correlations over train and test sets and weight vectors for CCA variants on a data set with a multivariate trigonometric relationship. Standard deviation is marked in parenthesis.

range from 0.26 to 0.30. The weight vectors indicate no success in discovering the relationship.

Experiment 8. Next, we define a trigonometric relationship between two first variables of each views as follows:

$$y_1 + y_2 = -x_1 + 2\sin(x_2) + 3\sin(x_2) + 4\sin(x_2) + \xi_1,$$

where x_1, x_2 and y_1 and y_2 are variables of views X and Y. The ground truth vectors are $u_g = [1\ 1\ 0\ 0\ 0\ 0]$, $v_g = [1\ 1\ 0\ 0\ 0\ 0]$.

The results in Figure 4.9 indicate that gradKCCA with a quadratic kernel manages to discover the relationship with an F1 score 0.77. GradKCCA with a Gaussian kernel also finds canonical weights close to the ground truth vectors with an F1 score 0.64. (S)CCA-HSIC has an F1 score 0.63, but there are incorrect weights on multiple variables and also the weight on x_2 is low (0.1).

Experiment 9. In this experiment, we create two non-monotonic relationships as follows:

$$y_1 = x_1^2 + x_2^2 + \xi_1$$

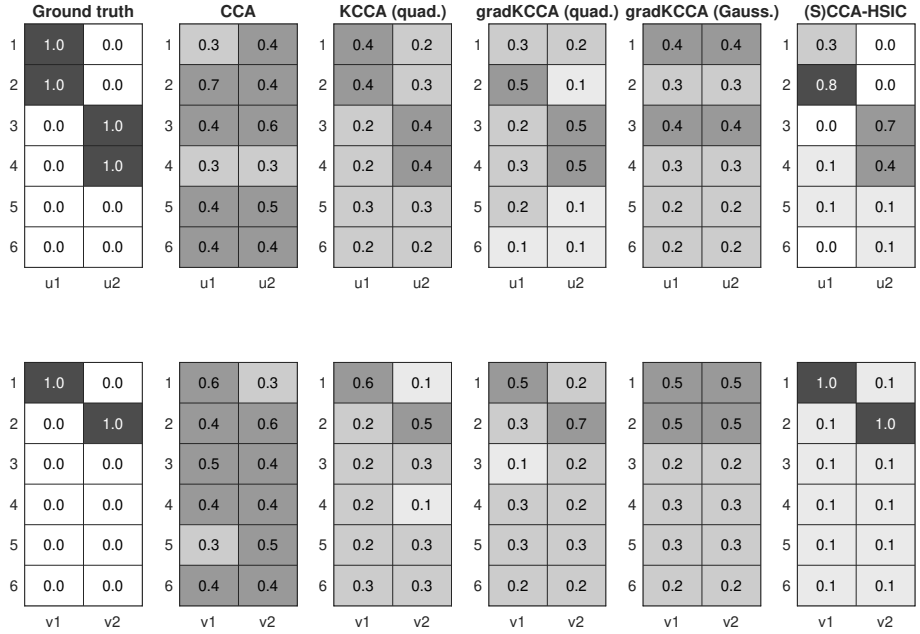
$$y_2 = x_3^4 + x_4^4 + \xi_2,$$

where y_1, y_2, x_1, x_2, x_3, and x_4 are variables of views X and Y, respectively. The ground truth vectors are $u_{g1} = [1\ 1\ 0\ 0\ 0\ 0]$, $v_{g1} = [1\ 0\ 0\ 0\ 0\ 0]$, $u_{g2} = [0\ 0\ 1\ 1\ 0\ 0]$, $v_{g2} = [0\ 1\ 0\ 0\ 0\ 0]$.

As Figure 4.10 shows, CCA, KCCA (quadratic kernel), gradKCCA (quadratic kernel), gradKCCA (Gaussian kernel) do not manage to discover the relationships. Instead, the F1 scores of the methods range from 0.37 to 0.42 and the methods put weight on incorrect variables. However, (S)CCA-HSIC is able to discover the relatioship with F1 score 0.52, R precision 0.73 and small incorrect weights.

	F1	R precision	ρ_{tr1}	ρ_{tr2}	ρ_{tt1}	ρ_{tt2}
CCA	0.38 (0.05)	0.42 (0.12)	0.24 (0.03)	0.16 (0.03)	0.06 (0.05)	0.05 (0.04)
KCCA (quad.)	0.37 (0.10)	0.47 (0.24)	0.58 (0.04)	0.49 (0.03)	0.28 (0.12)	0.09 (0.06)
gradKCCA (quad.)	0.42 (0.10)	0.52 (0.15)	0.65 (0.03)	0.50 (0.06)	0.55 (0.06)	0.44 (0.06)
gradKCCA (Gauss.)	0.38 (0.04)	0.40 (0.12)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)
(S)CCA-HSIC	0.53 (0.07)	0.73 (0.09)	0.57 (0.02)	0.57 (0.04)	0.54 (0.05)	0.55 (0.05)

(a) F1 score and correlations over train and test sets for CCA variants. Standard deviation is marked in parenthesis.



(b) Mean weight vectors over 10 runs of first and second components.

Figure 4.10: The results of CCA variants on data with two non-monotonic relationships.

	F1	R precision	ρ_{tr1}	ρ_{tt1}	$\tilde{u}$	$\tilde{v}$
CCA	1.00 (0.00)	1.00 (0.00)	0.97 (0.00)	0.97 (0.00)	[0.7 0.7 0 0 0 0]	[1.0 0.2 0 0 0 0]
KCCA (quad.)	0.64 (0.26)	0.55 (0.48)	0.87 (0.02)	0.82 (0.03)	[0.4 0.4 0.1 0.1 0.3 0.3]	[0.5 0.2 0.2 0.3 0.1 0.2]
gradKCCA (quad.)	0.79 (0.06)	1.00 (0.00)	0.90 (0.01)	0.88 (0.01)	[0.7 0.7 0 0 0 0]	[1 0.2 0 0 0 0]
gradKCCA (Gauss.)	0.73 (0.06)	1.00 (0.00)	0.90 (0.01)	0.88 (0.01)	[0.7 0.7 0 0 0 0.1]	[1 0.2 0 0.1 0 0.1]
(S)CCA-HSIC	0.99 (0.03)	1.00 (0.00)	0.93 (0.01)	0.93 (0.01)	[0.7 0.7 0 0 0 0]	[1 0.2 0 0 0 0]

Figure 4.11: F1 score, correlations over train and test sets and weight vectors for CCA variants on a data set with a multivariate exponential relationship.

Experiment 10. In this experiment, we test the capability of CCA variants to retrieve a multivariate exponential relationship. We define the variables as follows:

$$y_1 + y_2 = e^{x_1+x_2} + \xi_1,$$

where x_1, x_2 and y_1 and y_2 are respective variables of views X and Y. The ground truth vectors are $u_g = [1\ 1\ 0\ 0\ 0\ 0]$, $v_g = [1\ 1\ 0\ 0\ 0\ 0]$.

As is seen from Figure 4.11 there exists correlation between the variables which is the reason for CCA succeeding in finding the relationship with a perfect F1 score and R precision. (S)CCA-HSIC retrieves the relationship as well as CCA with F1 score 0.99. GradKCCA variants score 0.79 and 0.73, with a slightly better score and mean canonical weights for gradKCCA with a quadratic kernel. KCCA has a lot of misplaced weights, therefore the F1 score is 0.64.

4.1.3 Non-monotonic and monotonic relationships in the same experiment

Experiment 11. In this part, our goal is to evaluate the capability of compared CCA variants to retrieve relationships in a setting where there are multiple relationships, some of which are non-monotonic. We define the following relationships:

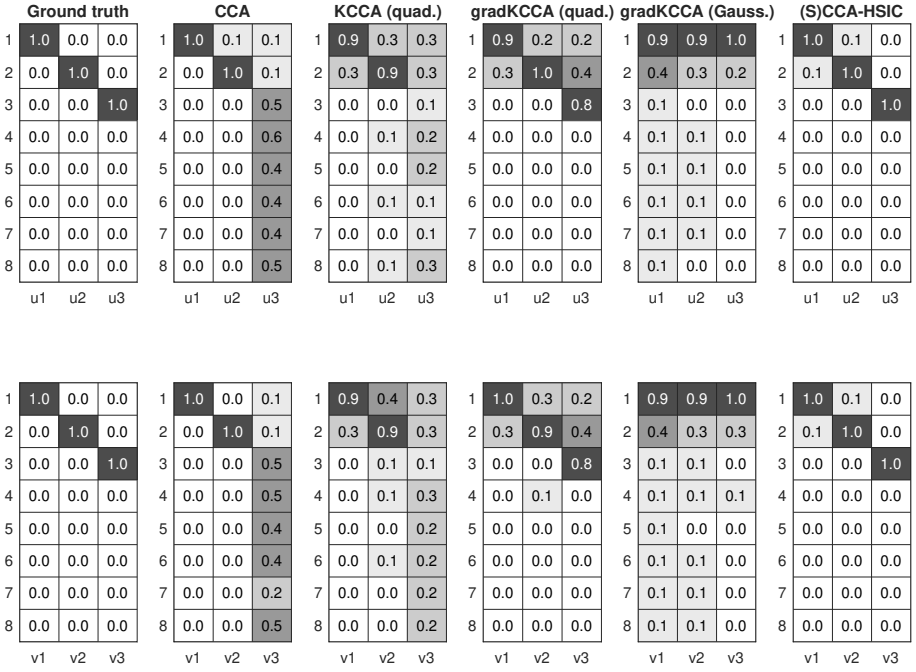
$$y_1 = x_1 + \xi_1$$
$$y_2 = x_2^3 + \xi_2$$
$$y_3 = x_3^2 + \xi_3,$$

where x_1 to x_3 and y_1 to y_3 are corresponding variables of views $X \in \mathbb{R}^{N\times p}$ and $Y \in \mathbb{R}^{N\times p}$ ($p = 8$, $q = 9$). The ground truth vectors are $u_{g1} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, $v_{g1} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, $u_{g2} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$, $v_{g2} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, $u_{g3} = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$, $v_{g3} = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$.

Figure 4.12 shows that CCA discovers the monotonic relationships but does not find the non-monotonic relationship, therefore scoring F1 of 0.68.

	F1	R precision	ρ_{tr1}	ρ_{tr2}	ρ_{tr3}	ρ_{tt1}	ρ_{tt2}	ρ_{tt3}
CCA	0.68 (0.07)	0.75 (0.12)	0.99 (0.00)	0.92 (0.00)	0.24 (0.04)	0.98 (0.00)	0.92 (0.01)	0.05 (0.04)
KCCA (quad.)	0.39 (0.07)	0.58 (0.14)	0.95 (0.00)	0.93 (0.01)	0.93 (0.01)	0.91 (0.01)	0.85 (0.03)	0.87 (0.01)
gradKCCA (quad.)	0.50 (0.10)	0.95 (0.11)	0.96 (0.01)	0.83 (0.07)	0.64 (0.14)	0.95 (0.01)	0.81 (0.08)	0.63 (0.13)
gradKCCA (Gauss.)	0.31 (0.01)	0.33 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
(S)CCA-HSIC	0.74 (0.13)	1.00 (0.00)	0.93 (0.02)	0.87 (0.02)	0.08 (0.03)	0.94 (0.02)	0.86 (0.03)	0.08 (0.05)

(a) F1 score and correlations over train and test sets for CCA variants.



(b) Mean weight vectors over 10 runs of first, second and third components.

Figure 4.12: The results of CCA variants on data with non-monotonic and monotonic relationships.

(S)CCA-HSIC has F1 score 0.74 and manages to discover all three relationships. GradKCCA (Gaussian kernel) puts weight on the first and second variables in every canonical weight pair, scoring 0.31. GradKCCA (quadratic kernel) does not discover the relationships separately like (S)CCA-HSIC does. Therefore, gradKCCA (quadratic kernel) scores 0.50. KCCA (quadratic kernel) scores 0.39 and is not capable of retrieving the third relationship.

4.2 Real-world data

In this chapter, we use four CCA variants, regular CCA, KCCA, gradKCCA and (S)CCA-HSIC on two real-world cases and compare the performance of the methods. The first data set is the body fat data set and the second data set is the Boston housing data set. The body fat data set is available at `http://www.statistics4u.com/fundstat_eng/data_bodyfat.html` and Boston data set is available at `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`.

4.2.1 Body fat data set

The body fat data set used in this experiment consists of 252 observations of American men, who were measured for the following 15 variables:

1. Density - Body density (g / cm^3)
2. %Fat - Body fat percentage
3. Age - Age of the person (year)
4. Weight - Weight of the person (lb)
5. Height - Height of the person (inch)
6. Neck - Neck diameter (cm)
7. Chest - Chest diameter (cm)
8. Abdomen - Abdomen diameter (cm)
9. Hip - Hip diameter in diameter (cm)
10. Thigh - Thigh diameter (cm)
11. Knee - Knee diameter (cm)
12. Ankle - Ankle diameter (cm)
13. Biceps - Biceps diameter (cm)
14. F-arm - Forearm diameter (cm)
15. Wrist - Wrist diameter (cm)

We divide the variables into two views. First view, X, consists of five variables: Density, %Fat, Age, Weight and Height. The second view, Y, consists of the remaining ten variables, which indicate the diameters of various body parts.

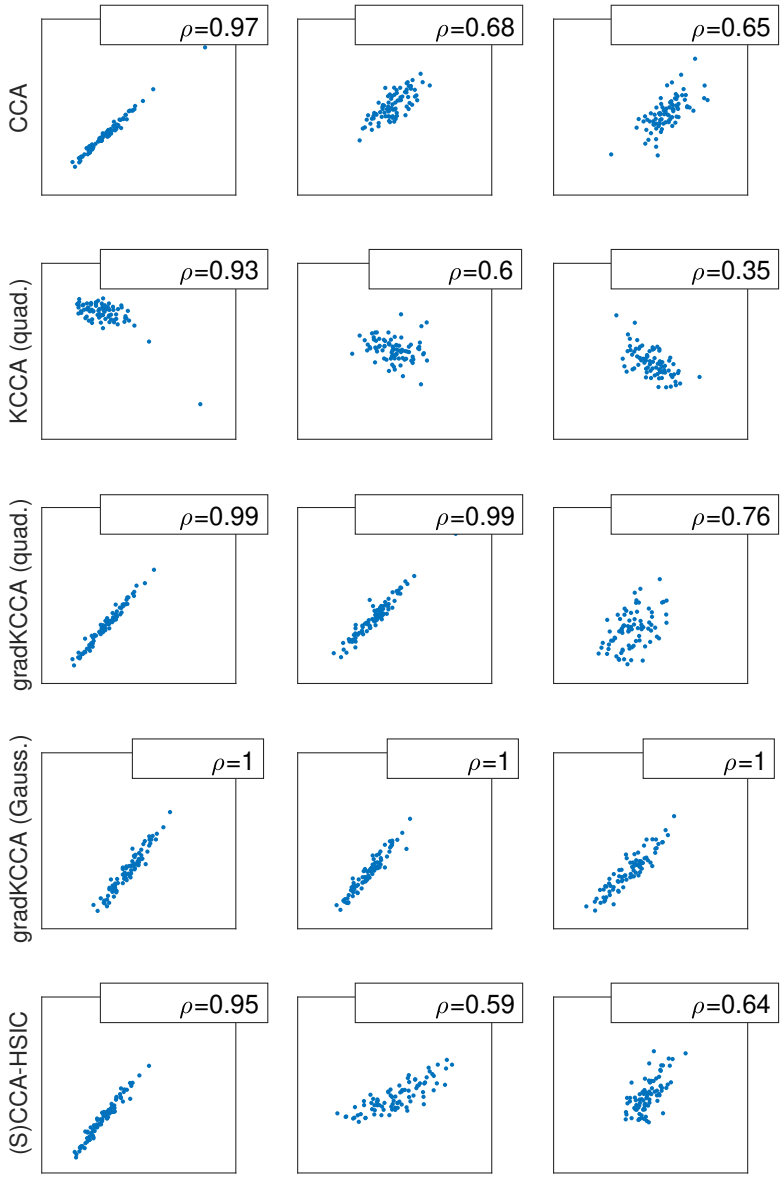
Figure 4.13 shows how all methods first find a relationship between Weight in the first view and multiple other variables in the second view. Regular CCA picks Chest, Abdomen and Hip as the variables with the most explanatory power in the second view. Non-linear methods differ in this sense. GradKCCA with a quadratic kernel puts weight on more variables than CCA, and so does gradKCCA with a Gaussian kernel and (S)CCA-HSIC. KCCA with a quadratic kernel puts weight on Hip, Neck and Forearm. Looking at how the points in the plots are on straight lines in Figure 4.13a, it can be concluded that all methods have found a linear relationship.

CCA finds two other relationships with rather weak correlations over test set, 0.68 and 0.65. In the second and third relationship the weights are on multiple variables, making it difficult to interpret without ignoring smaller weights. The second relationship has the highest coefficients on Age and Weight in the first view and Abdomen and Hip on the second view. The third canonical weight pair has the highest coefficients on Density and Age in the first view and Abdomen and Thigh in the second view.

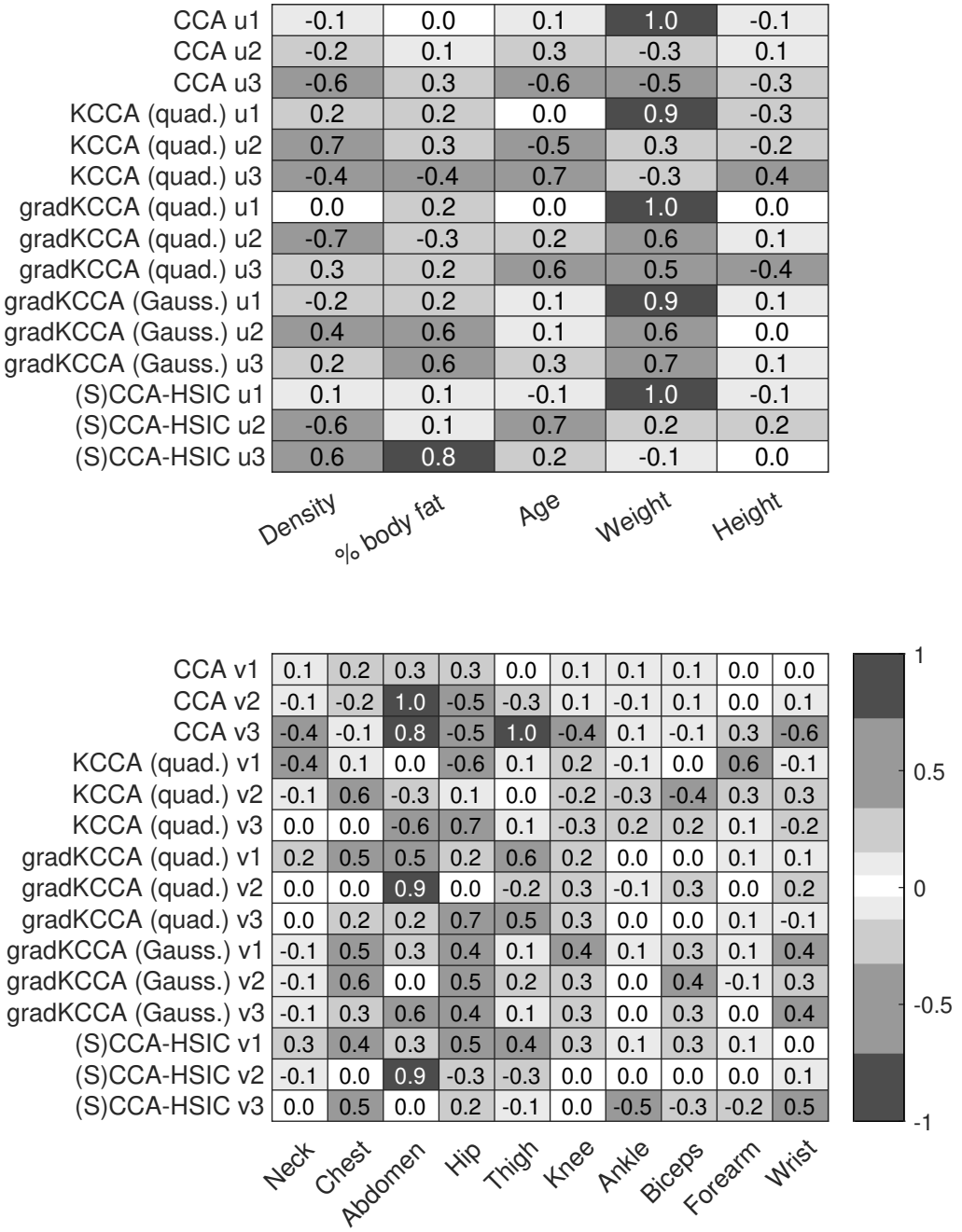
The second and third relationships of gradKCCA with a quadratic kernel are between multiple variables with varying signs and strengths. For the second relationship, there is a positive weight on Weight and a negative weight on Density in the first view, and in the second view the most dominant weight is on Abdomen. In the third canonical weight pair, there are high weights on all variables except Height, for which there is a negative weight. In the second view, Hip and Thigh have high weights. Despite interpreting the result being difficult, the plots of the first two relationships of gradKCCA in Figure 4.13a indicate that the relationships are linear and have high correlations.

The two other relationships that gradKCCA with a Gaussian kernel extracts are quite similar to each other. In both relationships, Weight, % body fat and Density have positive weights in the first view and all variables have similar weights in the second view, except Abdomen. For the second relationship, Abdomen has a 0.0 weight compared to 0.6 of the third relationship. Both relationships produce a linear relationship, as can be seen from Figure 4.13a.

The second relationship of (S)CCA-HSIC has weights of opposite signs on Age and Density in the first view, and weights on Abdomen, Hip and Tight in the second view. The third relationship mainly consists of positive weights on Density and % body fat in the first view, and a positive weight on Chest and Wrist and a negative weight on Ankle in the second view.



(a) Xu, Yv representation of the three canonical weight pairs for each compared method. In the upper right there is the corresponding canonical correlation over test set in the feature space for kernelized methods. For CCA, the value is the correlation over test set in the input space.



(b) The three canonical weight pairs of each CCA variant displayed as a heatmap.

Figure 4.13: The results of CCA variants on body fat data set. Shows the plots and canonical weights of the best run among 10 runs.

As Figure 4.13a shows, all relationships seem monotonic and linear. This could be due to the nature of the data set: there might not be any non-linear relationships in the data.

It can be noted that there are many non-zero values in the canonical weights. This may be due to the denseness of the compared methods. Having fewer weighted variables could help interpret the results better.

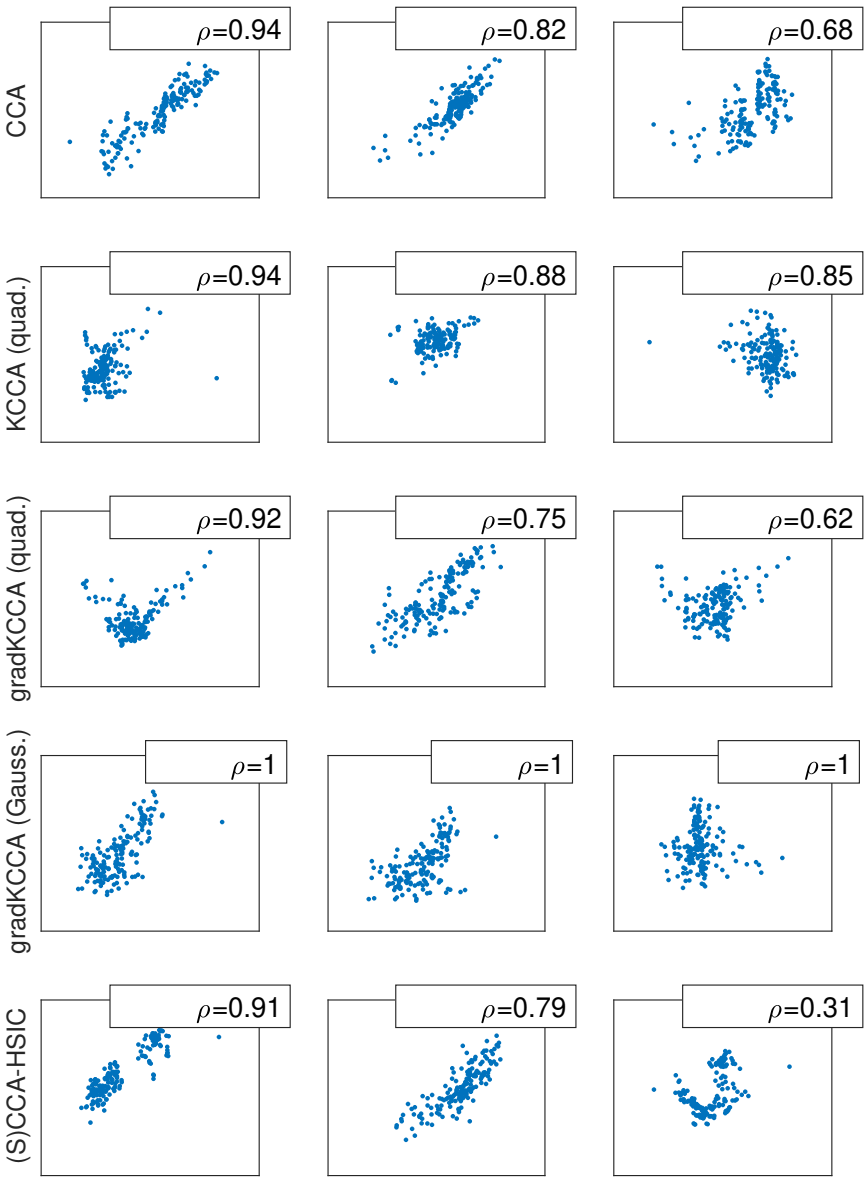
4.2.2 Boston housing data set

The Boston housing data set is a publicly available data set, which contains 506 observations of housing deals in the Boston area. The data consists of 14 variables as follows:

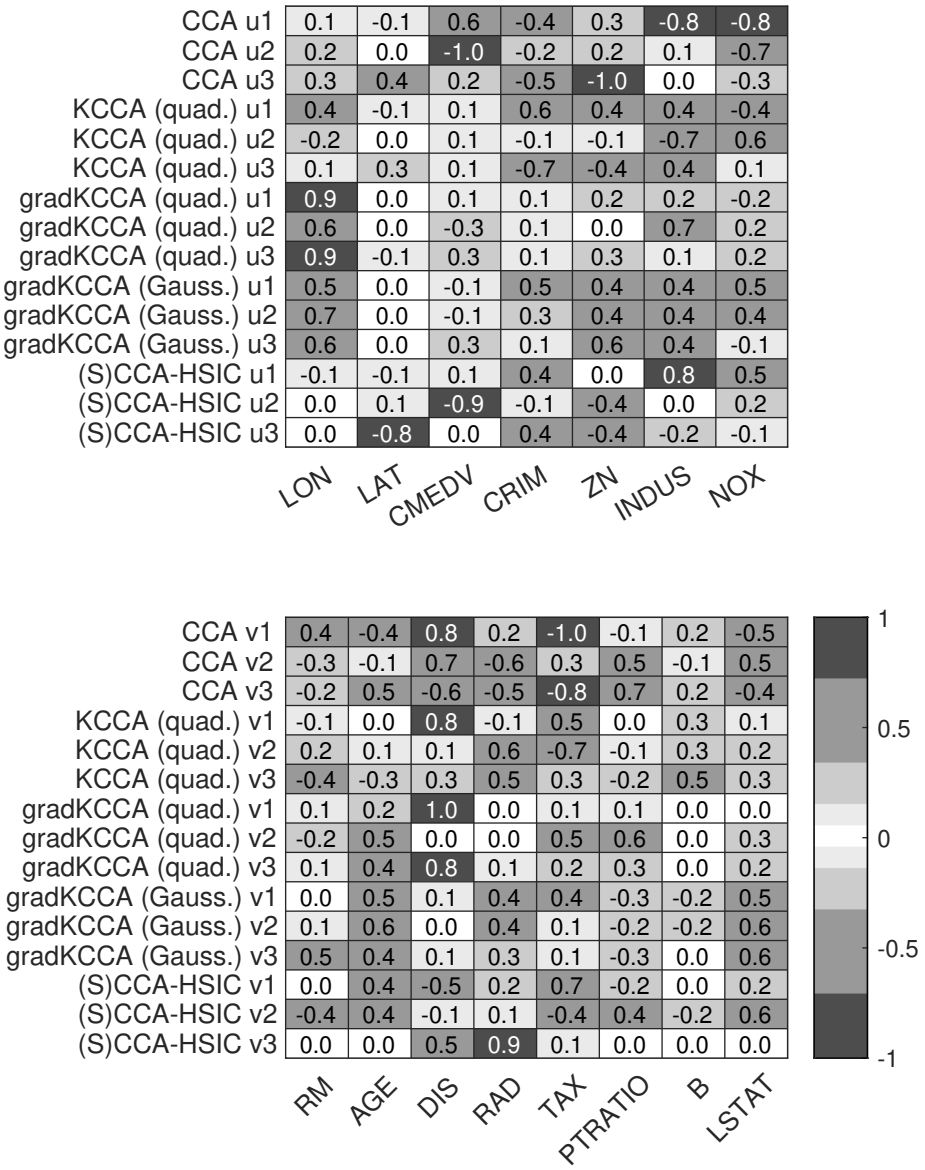
1. TRACT - Tract ID
2. LON - Longitude
3. LAT - Latitude
4. CMEDV - Median value of owner-occupied homes in $1000's
5. CRIM - Per capita crime rate by town
6. ZN - Proportion of residential land zoned for lots over 25,000 sq.ft.
7. INDUS - Proportion of non-retail business acres per town
8. NOX - Nitric oxides concentration (parts per 10 million)
9. RM - Average number of rooms per dwelling
10. AGE - Proportion of owner-occupied units built prior to 1940
11. DIS - Weighted distances to five Boston employment centres
12. RAD - Index of accessibility to radial highways
13. TAX - Full-value property-tax rate per $10,000
14. PTRATIO - Pupil-teacher ratio by town
15. B - $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
16. LSTAT - Percentage of lower status of the population

The first view, X, consists of variables LON, LAT, CMEDV, CRIM, ZN, INDUS, NOX. The second view, Y, consists of variables RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT. In this thesis, we leave variable TRACT out, since it does not provide any useful information.

The results of the compared methods can be seen in Figure 4.14. The first relationship that CCA finds can be most explained by INDUS, NOX and CMED in the first view and DIS, TAX in the second view. The large number of weighted variables makes it difficult to interpret the result. The found relationship has a high correlation over test set, 0.94. The second relationship mainly consists of CMEDV and NOX in the first view and DIS,



(a) Xu, Yv representation of the three canonical weight pairs for each compared method. In the upper right there is the corresponding canonical correlation over test set in the feature space for kernelized methods. For CCA, the value is the correlation over test set in input space.



(b) The three canonical weight pairs of each CCA variant displayed as a heatmap.

Figure 4.14: The results of CCA variants on Boston housing data set. Shows the plots and canonical weights of the best run among 10 runs.

RAD and various other variables in the second view, again making it hard to interpret. The third relationship is mostly explained by ZN in the first view, but has weights on all variables in the second view. The correlation over test set is rather low, 0.68.

KCCA with a quadratic kernel finds a relationship explained by CRIM, ZN, INDUS, NOX and LON in the first view and DIS, TAX and B in the second view. The second relationship has weights on LON, INDUS, CMEDV and NOX in the first view and TAX, AGE, PTRATIO and LSTAT in the second view. The third relationship can be mostly explained by LON in the first view and DIS in the second view. The plots of the relationships retrieved by KCCA do not seem to have high dependence, as is seen in Figure 4.14a.

The first canonical weight pair of GradKCCA with a quadratic kernel represents a relationship between DIS and LON with a high correlation in the feature space. Looking at the plot, it resembles a non-linear, non-monotonic relationship, since the Xu value starts high, then drops and then becomes high again. The plot of the second relationship looks linear, and it is mainly between LON and INDUS in the first view and AGE, TAX and PTRATIO in the second view. The canonical variables of the third relationship have a rather low correlation over test set in the feature space, 0.62, and is mainly produced by weights on variables LON and DIS.

GradKCCA with a Gaussian kernel finds three relationships with almost the same weights on each canonical weight pair. The weighted variables are LON, ZN, INDUS, NOX in the first view and AGE, RAD and LSTAT in the second view. Some differences in weights are on variables CRIM in the first view and RM and TAX in the second view. The high number of variables makes the interpretation difficult. The plot of the first relationship looks rather linear, whereas the second plot could possibly represent a weak, non-linear relationship. All correlations over test set are 1.0.

The first relationship of (S)CCA-HSIC puts weights primarily on INDUS, NOX and CRIM in the first view and TAX, DIS and AGE in the second view. Again, the large number of variables on both views makes it hard to interpret. The plot has two clusters and a space in between them, which could indicate a non-linear relationship. The second relationship is the known non-linear relationship between CMEDV and LSTAT [2]. The third canonical weights represent a relationship primarily between LAT and RAD with opposite signs. The plot has a peculiar shape and could have dependence.

Chapter 5

Discussion

In this thesis, our goal was to empirically evaluate the performances of the four CCA methods, CCA, KCCA, gradKCCA and (S)CCA-HSIC in linear and non-linear settings using simulated and real-world scenarios. We ran 12 experiments on simulated data, aiming to find out what sort of relationships each method is capable of discovering and where the limits are for the complexity of the relationships. In real-world experiments, we used two publicly available data sets to learn what kind of relationships each method finds in real-world settings.

The orthogonality of the canonical weights did not hold in all experiments. In Experiment 11, we had one linear, one cubic and one quadratic relationship. There, gradKCCA (Gaussian kernel) seems to find the same relationship multiple times by returning three almost identical canonical weight pairs. Also, with the Boston housing data, gradKCCA (Gaussian) appeared to return three almost identical canonical weight pairs. With CCA for example, returning three same canonical weight pairs should not be possible, since the orthogonality of the solutions is guaranteed when solving the eigenvalues in closed form. The part of solving gradKCCA that is responsible for orthogonality of the canonical weights is the matrix deflation in the numerical gradient based algorithm (Algorithm 1, line 28). The experiment results indicate that there might be problems with matrix deflation for gradKCCA. If that is the case, the matrix deflation method needs to be improved to make the non-linear methods retrieve multiple relationships as well as CCA.

The uncovering of the relationships could depend on the number of observations. In Experiment 6.1, we had a long trigonometric relationship, and in Experiment 7, we had a circle shaped relationship. In both experiments, none of the compared methods could discover the relationship. We continued Experiment 6.1 in Experiment 6.2 by keeping the relationship the same but increasing the number of observations. It turned out that when the number

of observations was increased, the methods would slowly begin to recognize the relationship. Therefore, it could be that the relationship was not inherently too complex for the methods. The experiment might suggest that, if recognizing the relationship is too difficult for the method, more observations may be needed to counteract the difficulty. However, even if increasing the number of observations improved the capability of CCA variants to recognize non-linear relationships, that does not imply that all complex relationships could be recognized that way. There could still exist inherently too complex relationships for the used methods regardless of the number of observations. For example, for CCA, any non-monotonic relationship is too complex, regardless of the number of observations.

In Experiment 6.1, it is also interesting that none of the used methods recognized the relationship with only 500 observations. If we take the plot in Figure 4.6a and ask a person to answer the question "When x_1 is 0.2, what value will y_1 likely be?", the person would recognize the dependency between x_1 and y_1 and easily tell only from the 500 observations that y_1 will likely get a value close to one. Therefore, in theory, the dependency should be recognizable with only 500 observations. In machine learning, the need for a large number of observations is a well-recognized problem for many machine learning algorithms.

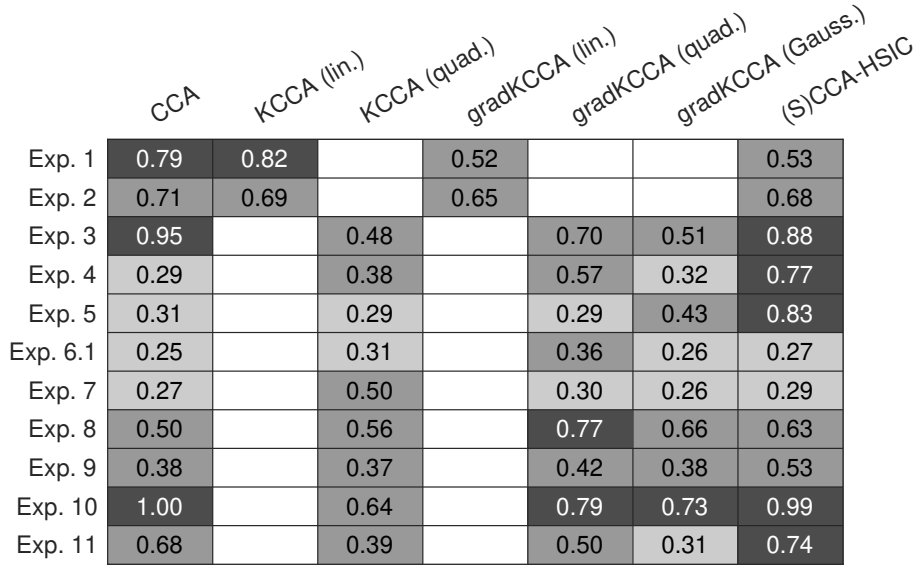
Experiment 3 showed that CCA could obtain better scores for a non-linear, monotonic relationship than the non-linear CCA variants. In the experiment, we had a single cubic relationship in the data set. The highest possible multilinear correlation that can be found in the data should be given by the ground truth vectors, in other words as a bivariate correlation between x_1 and y_1. The theoretical correlation is roughly the correlation over test set of CCA, 0.91. A perfect modeling could be expected to have a correlation of 1.0 over test set value of 1.0 (when the noise is low), like in Experiment 1 for CCA. Since CCA is a linear method, it models the cubic relationship with a linear model, which is an inaccurate model for an x^3 shaped relationship. Therefore, it is interesting that the non-linear variants, which should have the potential to use more complicated modeling for the relationship than CCA, seem to perform worse with regard to the F1 score and R precision than CCA. Perhaps, the slight errors in the canonical weights that the non-linear methods retrieve are due to some amount of overfitting, which is known to be more likely for non-linear methods. Then, the metrics penalize the non-linear methods for the overfitting, which outweighs the improvements of the better modeling of the relationship.

In Experiment 1, CCA separated the three linear relationships as expected, but in Experiment 2 with two multilinear relationships CCA partly mixed the two relationships. The lack of separation of relationships for CCA

does not mean that the result is necessarily wrong. Either way, CCA has found the best coefficients to maximize the correlation between the canonical variables with the given the data set. There is no guarantee that the highest correlation will be found when the relationships are fully separated. For example, if we define $x_1 = y_1$ and $x_2 = y_2$ without any noise, the correlation between x_1 and y_1 is 1.0 (also for x_2 and y_2). However, the correlation between $\alpha x_1 + \beta x_2$ and $\alpha y_1 + \beta y_2$ is also 1.0 ($-1 \leq \alpha \leq 1$, $-1 \leq \beta \leq 1$), so either solution could be chosen as the solution for CCA. The intuition becomes more complicated with noise added. The noise creates randomness, which perhaps causes one solution to have a higher correlation over the other, therefore causing CCA to sometimes separate the relationships (Experiment 1) and other times not (Experiment 2). Despite both answers being technically acceptable, for a researcher studying an unknown data, it is probably more beneficial to have the relationships separated as atomically as possible.

In the experiments of this thesis, we used multiple different metrics for evaluating the results. Many of the metrics have their shortcomings, but having several metrics should help making the right conclusion. For the most part, F1 score was used to evaluate the overall performance. One advantage of F1 score is that it is a single value and easily comparable to the results of other experiments. One disadvantage is that the threshold for determining the incorrect weights needs to be set, and the choice is arbitrary. Also, the general problem with step functions is that a small deviation in the input can result in a completely different output. However, an advantage that F1 has over R precision, which has no parameters, is that F1 score seems to create more difference between the compared methods. In Experiment 3 for example, the R precisions of gradKCCA (both Gaussian and quadratic) and (S)CCA-HSIC were 1.0, although there was a clear difference in the retrieved canonical weights in favor of (S)CCA-HSIC. F1 score reflects this difference with the highest F1 score for (S)CCA-HSIC.

The mean canonical weights are method-specific metrics used in the experiments. They provide exact and essential low-level information about the performances of the methods. However, one needs to be careful when using the mean. Taking the mean loses lots of information about the separate runs, and some of the lost information could potentially be important. Using mean might also result in a situation where none of the canonical weights of the separate runs resemble the mean canonical weights, which could mislead in their interpretation. Mean canonical weights do not work as an overall scoring metric because the correct weights always require knowledge of the experiment setup, whereas interpreting F1 score or R precision can be done without knowledge of the experiment setup.



	CCA	KCCA (lin.)	KCCA (quad.)	gradKCCA (lin.)	gradKCCA (quad.)	gradKCCA (Gauss.)	(S)CCA-HSIC
Exp. 1	0.79	0.82		0.52			0.53
Exp. 2	0.71	0.69		0.65			0.68
Exp. 3	0.95		0.48		0.70	0.51	0.88
Exp. 4	0.29		0.38		0.57	0.32	0.77
Exp. 5	0.31		0.29		0.29	0.43	0.83
Exp. 6.1	0.25		0.31		0.36	0.26	0.27
Exp. 7	0.27		0.50		0.30	0.26	0.29
Exp. 8	0.50		0.56		0.77	0.66	0.63
Exp. 9	0.38		0.37		0.42	0.38	0.53
Exp. 10	1.00		0.64		0.79	0.73	0.99
Exp. 11	0.68		0.39		0.50	0.31	0.74

Figure 5.1: Summary of F1 scores of all variants for 11 simulated experiments.

The results regarding CCA were mostly expected. In experiments with linear and monotonic relationship, CCA found the relationships well. In Experiment 2, CCA mixed together the two linear relationships, which was not expected. CCA could not retrieve non-monotonic relationships in any experiment, as expected. In both real-world experiments, CCA found relationships with high correlation over the test set.

KCCA worked comparably well with a linear kernel, retrieving a result similar to CCA in Experiment 1 and performing similarly to kernelized variants in Experiment 2 (see Figure 5.1). In Experiments 3, 4, 5, 8, 10 and 11, KCCA with a quadratic kernel had trouble finding the relationship without error in weights while some or all other kernelized methods could find the relationship. KCCA with a quadratic kernel did not perform the best in any of the experiments. The performance of KCCA might have been negatively affected by using only quadratic kernels, instead of including Gaussian kernels in the experiments. Also, using matrix decomposition approaches such as the incomplete Cholesky decomposition could potentially improve the results for KCCA.

GradKCCA was able to retrieve the relationship in Experiments 1, 2, 3, 4, 5, 6.1, 8, 10, 11 but lost to (S)CCA-HSIC in terms of quality of the canonical weights in Experiments 3, 4, 10, 11 (see Figure 5.1). It performed better than (S)CCA-HSIC in Experiment 8 using a quadratic kernel. GradKCCA with a quadratic kernel performed better than gradKCCA with a Gaussian kernel in Experiments 3, 4, 11. In Experiment 5, gradKCCA with a Gaussian kernel

performed better than gradKCCA with a quadratic kernel. GradKCCA with a Gaussian kernel had more noisy canonical weights than gradKCCA with a quadratic kernel in Experiments 3 and 4 and retrieved incorrect canonical weights in Experiment 11. While Experiment 6.2 does suggest that increasing the amount of observations could improve the performance of gradKCCA with a Gaussian kernel, it does not seem that more observations would make it work better than gradKCCA with a quadratic kernel. Perhaps the performance and the extent of possible overfitting of gradKCCA with a Gaussian kernel could be improved by optimizing the bandwidth parameter used in the kernel, instead of using the median trick.

As is seen in Figure 5.1, (S)CCA-HSIC yielded the best F1 score in Experiments 4, 5, 9, 10. However, (S)CCA-HSIC did not manage to find the relationship in Experiments 6.1, 7, 8. In real-world data, (S)CCA-HSIC managed to find relationships well.

5.1 Conclusion

In this thesis, we studied CCA and three kernelized CCA variants, KCCA, gradKCCA and (S)CCA-HSIC, in 12 simulated experiments and two experiments with real-world data. In the simulated experiments, we found out that (S)CCA-HSIC was able to retrieve the relationships with the most consistency and highest resemblance to the ground truth vectors out of the compared methods. GradKCCA and KCCA also showed capability to discover complex, non-linear relationships. Selection of the gradKCCA kernel could have a significant effect on the performance of the method. However, the results could potentially change by choosing the experiments differently. For example, the number of observations in the experiments was relatively low. Regarding KCCA, it could be represented with a Gaussian kernel, and some sort of matrix decomposition could be used. Also, it would be interesting to see how using sparse versions of gradKCCA and (S)CCA-HSIC affects the amount of noise in the canonical weights and overall capability to discover complex, non-linear relationships. In future work, more thorough experimenting will be needed on the subject.

In the real-world experiments of this thesis, we compared the four methods with two real-world data sets. In the experiments, all compared methods showcased capabilities to discover linear relationships in the data. GradKCCA and (S)CCA-HSIC showed hints of capability to discover non-linear relationships in real-world data, but no conclusive results were found regarding dense KCCA, gradKCCA and (S)CCA-HSIC. The canonical weights had too many variables with a non-zero weight for an easy and intuitive interpre-

tation. Using sparse methods could uncover more easily interpretable canonical weights, which could be investigated in future work. Also, different data sets with more non-linear relationships may be needed to further experiment the capabilities of the kernelized variants to find non-linear relationships in real-world settings.

Bibliography

[1] Akaho, S., 2001. A kernel method for canonical correlation analysis. *Proceedings of the International Meeting of the Psychometric Society (IMPS'01)*.

[2] Chang, B. et al., 2013. Canonical correlation analysis based on Hilbert-Schmidt independence criterion and centered kernel target alignment. *International Conference on Machine Learning*, pp. 316–324.

[3] Uurtio, V., Bhadra, S., Rousu, J., 2018. Sparse non-linear CCA through Hilbert-Schmidt independence criterion. *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1278–1283, IEEE.

[4] Hotelling, H., 1936. Relations between two sets of variates. *Biometrika*, vol. 28, no. 3/4, pp. 321–377.

[5] Gretton, A. et al., 2005. Measuring statistical dependence with Hilbert-Schmidt norms. *Algorithmic Learning Theory: 16th International Conference*, ALT 2005, pp. 63–78.

[6] Uurtio, V., Bhadra, S., J. Rousu, 2019. Large-scale sparse kernel canonical correlation analysis. *International Conference on Machine Learning*, pp. 6383–6391.

[7] Isotalo, J., 2001. *Basics of statistics*. Finland: University of Tampere.

[8] Bolstad, W. M., 2007. *Introduction to Bayesian Statistics*. Wiley-Interscience.

[9] Galton, F., 1885. Inheritance and regression.

[10] Pearson, K., 1896. Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London*, Series A, Containing Papers of a Mathematical or Physical Character, vol. 187, pp. 253–318.

[11] Lee Rodgers, J., Nicewander, W.A., 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician* 42, no. 1, pp. 59–66.

[12] Mari, D. D., Kotz, S., 2001. *Correlation and Dependence*. World Scientific.

[13] Uurtio, V. et al., 2018. A tutorial on canonical correlation methods. *ACM Computing Surveys (CSUR)*, 50, no. 6, p. 95.

[14] Higgins, J. P., 2002. Nonlinear systems in medicine. *The Yale journal of biology and medicine*, 75, no. 5-6, p. 247.

[15] Hotelling, H., 1935. The most predictable criterion. *Journal of Educational Psychology*, 26, no. 2, p. 139.

[16] Leurgans, S. E., Moyeed, R. A., Silverman, B. W., 1993. Canonical correlation analysis when the data are curves.*Journal of the Royal Statistical Society: Series B (Methodological)*, 55, no. 3, pp. 725–740.

[17] Thompson, B., 1991. A primer on the logic and use of canonical correlation analysis. *Measurement and Evaluation in Counseling and Development*.

[18] Sherry, A., Henson, R. K., 2005. Conducting and interpreting canonical correlation analysis in personality research: A user-friendly primer. *Journal of Personality Assessment*, 84, no. 1, pp. 37–48.

[19] Melzer, T., Reiter, M., Bischof, H., 2001. Nonlinear feature extraction using generalized canonical correlation analysis. *International Conference on Artificial Neural Networks*, pp. 354–360, Springer, Berlin, Heidelberg.

[20] Rojo-Alvarez, J. L. et al., 2017. *Digital Signal Processing with Kernel Methods*. Hoboken: Wiley.

[21] Fyfe C., Lai, P. L., 2000. Canonical correlation analysis neural networks. *Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, pp. 997–980, IEEE.

[22] Van Gestel, T., et al., 2001. Kernel canonical correlation analysis and least-squares support vector machines. *International Conference on Artificial Neural Networks*, pp. 384–389, Springer, Berlin, Heidelberg.

[23] Bach, F. R., Jordan, M. I., 2002. Kernel independent component analysis. *Journal of Machine Learning Research*, 3, July, pp. 1–48.

[24] Hardoon, D. R., Szedmak, S., Shawe-Taylor, J., 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16, no. 12, pp. 2639–2664.

[25] Meila, M., 2003. Data centering in feature space. *AISTATS*.

[26] Olsson, D., 2011. Applications and implementation of kernel principal component analysis to specific data sets. Master's thesis. University of Florida.

[27] Lopez-Paz, et al., 2014. Randomized non-linear component analysis. *International Conference on Machine Learning*, pp. 1359–1367.

[28] Wang, W., Livescu, K., 2015. Large-scale approximate kernel canonical correlation analysis. arXiv preprint arXiv:1511.04773.

[29] Yoshida, K., Yoshimoto, J., Doya, K., 2017. Sparse kernel canonical correlation analysis for discovery of non-linear interactions in high-dimensional data. *BMC bioinformatics*, 18, no. 1, p. 108.

[30] Rahimi, A., Recht, B, 2008. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, pp. 1177–1184.

[31] Krstajic, D., et al., 2014. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 6, no. 1, p. 10.